Robust relationship extraction in the biomedical domain

D I S S E R T A T I O N

zur Erlangung des akademischen Grades

Dr. rer. nat.
im Fach Informatik

eingereicht an der
Mathematisch-Naturwissenschaftlichen Fakultät
Humboldt-Universität zu Berlin

von
M.Sc. Philippe Thomas

Präsident der Humboldt-Universität zu Berlin:
Prof. Dr. Jan-Hendrik Olbertz

Dekan der Mathematisch-Naturwissenschaftlichen Fakultät:
Prof. Dr. Elmar Kulke

Gutachter:
1. Ulf Leser
2. Kevin Bretonnel Cohen
3. Pierre Zweigenbaum

eingereicht am: 25.2.2015
Tag der mündlichen Prüfung: 13.7.2015

Abstract

For several centuries, a great wealth of human knowledge has been communicated by natural language, often recorded in written documents. In the life sciences, an exponential increase of scientific articles has been observed, hindering the effective and fast reconciliation of previous finding into current research projects. Many of these documents are freely provided in computer readable formats, enabling the automatic extraction of structured information from unstructured text using text mining techniques. This thesis studies a central problem in information extraction, *i.e.,* the automatic extraction of relationships between named entities. Within this topic, it focuses on increasing robustness for relationship extraction, which is analyzed in three different schemes:

First, we evaluate the use of ensemble methods to improve performance using data provided by the drug-drug-interaction challenge 2013. Ensemble methods aggregate several classifiers into one model, increasing robustness by reducing the risk of choosing an inappropriate single classifier. We will show that ensemble methods achieve a higher performance compared to individual classifiers on hidden test data.

Second, this work discusses the problem of applying relationship extraction to documents with unknown text characteristics. Corpora are usually sampled from a large text collection using some formal criterion and therefore often reflect only a specific subdomain. This affects performance of learned text mining components on texts with potentially different properties. Robustness of a text mining component is assessed by cross-learning, where a model is evaluated on a corpus different from the training corpus. We apply self-training, a semi-supervised learning technique, in order to increase cross-learning performance and show that it is more robust in comparison to a classifier trained on manually annotated text only.

Third, we investigate the use of distant supervision to overcome the need of manually annotated training instances. Corpora derived by distant supervision are inherently noisy, thus benefiting from robust relationship extraction methods. We compare two different methods capable of learning from distantly labeled corpora. The first method uses a state-of-the-art machine learning algorithm to learn a statistical model. Second, we learn patterns from positive instances. Both approaches achieve similar performance as fully supervised classifiers, evaluated in the cross-learning scenario.

To facilitate the usage of information extraction results, including those developed within this thesis, we develop the semantic search engine GeneView. GeneView is built upon a comprehensively annotated version of MEDLINE citations and openly available PubMed Central full texts. We discuss computational requirements to build this resource and present some applications utilizing the data extracted by different text-mining components.

Zusammenfassung

Schon seit Jahrhunderten wird menschliches Wissen in Form von natürlicher Sprache ausgetauscht und in Dokumenten schriftlich aufgezeichnet. In den letzten Jahren konnte man auf dem Gebiet der Lebenswissenschaften eine exponentielle Zunahme wissenschaftlicher Publikationen beobachten. Ein effektiver und schneller Zugriff auf frühere Erkenntnisse für die aktuelle Forschungsarbeit ist somit nur schwer umsetzbar. Die Extraktion relevanter Informationen in strukturierter Form kann mit Hilfe von Textmining-Methoden aus unstrukturierten Texten ermöglicht werden, sofern diese in computerlesbarem Format vorliegen. Diese Dissertation untersucht ein zentrales Problem der Informationsextraktion, nämlich die automatische Extraktion von Beziehungen zwischen Eigennamen. Innerhalb dieses Gebietes beschäftigt sich die Arbeit mit der Steigerung der Robustheit für die Relationsextraktion. Diese wird in drei verschiedenen Systemen untersucht.

Zunächst wird der Einsatz von Ensemble-Methoden anhand von Daten aus der "Drug-drug-interaction challenge 2013" evaluiert. Ensemble-Methoden erhöhen die Robustheit durch Aggregation unterschiedlicher Klassifikationssysteme zu einem Modell. Dadurch verringert sich das Risiko der Wahl eines unpassenden Klassifikators. Es wird gezeigt, dass Ensemble-Methoden eine bessere Leistung erzielen als die Verwendung einzelner Klassifikatoren.

Weiterhin wird in dieser Arbeit das Problem der Relationsextraktion auf Dokumenten mit unbekannten Texteigenschaften beschrieben. Annotierte Korpora spiegeln oft nur eine bestimmte Sub-Domäne wieder, da sie in der Regel mit formalen Kriterien aus großen Textsammlungen erstellt werden. Dies beeinträchtigt die Leistung darauf erlernter Text-Mining-Komponenten bei Korpora, welche abweichende Charakteristiken im Vergleich zum Trainingskorpus besitzen. Es wird gezeigt, dass die Verwendung des halb-überwachten Lernverfahrens self training in solchen Fällen eine höhere Robustheit erzielt als die Nutzung eines Klassifikators, der lediglich auf einem manuell annotierten Korpus trainiert wurde. Zur Ermittlung der Robustheit wird das Verfahren des cross-learnings verwendet. Dieses Verfahren beurteilt ein Modell auf einem vom Trainingskorpus abweichenden Korpus. Durch die Anwendung der Methode des self training wird die cross-learning Leistung deutlich verbessert.

Zuletzt wird die Verwendung von distant-supervision untersucht. Mit Hilfe dieses Verfahrens wird die Notwendigkeit von manuell annotierten Trainingsinstanzen überwunden. Korpora, welche mit der distant-supervision-Methode erzeugt wurden, weisen ein inhärentes Rauschen auf und profitieren daher von robusten Relationsextraktionsverfahren. Es werden zwei verschiedene Methoden untersucht, die auf solchen Korpora trainiert werden. Das erste Verfahren verwendet einen modernen Maschinenlernalgorithmus um ein statistisches Modell zu erlernen. Bei der zweiten Methode werden Graphmuster aus positiv markierten Trainingsinstanzen erlernt. Beide Ansätze zeigen eine vergleichbare Leistung wie vollständig überwachte Klassifikatoren, welche mit dem cross-learning-Verfahren evaluiert wurden.

Um die Nutzung von Ergebnissen der Informationsextraktion zu erleichtern, wurde die semantische Suchmaschine GeneView entwickelt. GeneView basiert auf einer umfassend annotierten Version von MEDLINE und öffentlich zugänglichen Volltexten aus PubMed-Central. Anforderungen an die Rechenkapazität beim Erstellen von GeneView werden diskutiert und Anwendungen auf den von verschiedenen Text-Mining-Komponenten extrahierten Daten präsentiert.

Acknowledgement

This PhD thesis would not have been possible without the encouragement, assistance, and support of many people. First, I would like to express my sincere gratitude to my supervisor, Prof. Ulf Leser, for his continuous support and for giving me the opportunity to work at his research group. He always encouraged me to tackle research problems from different perspectives and to ask critical questions. I also would like to thank Roman Klinger, who supervised my master thesis and developed my interest in the topic of natural language processing.

My colleagues of the WBI group have largely influenced my personal and professional development in the last years. I will miss the atmosphere and the strong social cohesion. I would like to thank Tim Rocktäschel and Michael Weidlich for numerous discussions; Karin Zimmermann for advise on statistical methods and for providing me with tasty apples; Astrid Rheinländer and Stefan Kröger for help on algorithmic questions and interesting insights into everyday matters; Marc Bux for giving inspiring presentations about his research, showing me the culinary side of Berlin, and the organization of many social activities; Berit Haldemann for profound discussions and the joint exploration of Berlin; Illés Solt and Domonkos Tikk for a great scientific collaboration and a guided tour through Budapest. Furthermore, I would like to thank André Koschmieder, Johannes Starlinger, Liam Childs, Sebastian Arzt, Lars Döhling, and Mariana Neves for their support and for interesting conversations during lunch time.

I also would like to express my gratitude to those people I met while working at University of Colorado Denver. Especially, Haibin Liu, Kevin Cohen, Larry Hunter, Mike Bada, Bill Baumgartner, Chris Roeder, Kevin Livingston, and Carsten Görg. Many thanks to Elaine Epperson, who accompanied me to many events and showed me several places in and around Denver.

I would also like to thank Rong Chen who gave me the opportunity to join his group as a short-term researcher at the Mount Sinai Hospital. I am very grateful to Jörg Hakenberg who helped me organizing my visit and for many fruitful discussions on our joint project.

Finally, I want to thank my family for their constant care, support, and unconditional love. I consider myself lucky having so many great people in my far and close proximity.

Contents

1 Introduction

1.1 Motivation

In early days, scientists communicated findings with other researchers by writing letters. For instance, Johannes Kepler wrote several letters to his contemporary Galileo Galilei to discuss heliocentricism and the discovery of Jupiter's satellites. Later, scientific journals enabled researchers to reach a broader community compared to direct communication by hand written letters. For several years journal articles are stored and collected electronically. MEDLINE contains the largest digital collection of biomedical articles with publications dating back to the early 19th century[1].

This repository already contains more than 24 million citations with a fast increasing number of articles. In fact, more than 50 % of all archived articles have been published within the last 19 years. Hunter and Cohen (2006) estimated a double-exponential increase of biomedical literature for the years 1986 to 2005[2]. The MEDLINE repository covers only bibliographic information, such as title, abstract, authors, and journal. To this end, the National Library of Medicine collects full-text articles in a repository called PubMed Central (PMC). Currently, this repository covers more than 3.3 million full-texts from more than 1,600 participating journals. Most journals currently restrict access to human readers and prohibit automatic text-analysis for any purpose. Hence, approximately 700 k full-texts are currently available for automatic text analysis. The annual increase of new MEDLINE citations and PMC (open access) full-texts is shown in Figure 1.1.

These numbers point towards a problem modern researchers are facing: The amount of published information is beyond the ability of researchers to grasp every detail on their research topic. PubMed, the most widely used interface to access MEDLINE, retrieves 291,460 articles when searching for "human immunodeficiency virus"[3]. Hence, ranking of relevant articles is a tough problem. PubMed currently ranks articles by indexing date.

Researchers often search relevant information in specialized structured databases. For instance, a researcher interested in all interaction partners of TP53 would probably first look into databases such as UniProtKB (Magrane and Uniprot Consortium, 2011) or IntAct (Aranda *et al.*, 2010) before searching for relevant articles. It would be extremely time consuming to find publications describing "all" known interaction partners of TP53. However, novel research findings are usually articulated in scientific articles

[1] The oldest electronically available publications are published in the journal of "Medico-Chirurgical Transactions" dating back to 1809. (*e.g.,* PMID 20895125)

[2] Coefficient of determination for linear regression was estimated for the individual years with $R^2 = 0.95$.

[3] As of 02/09/2015.

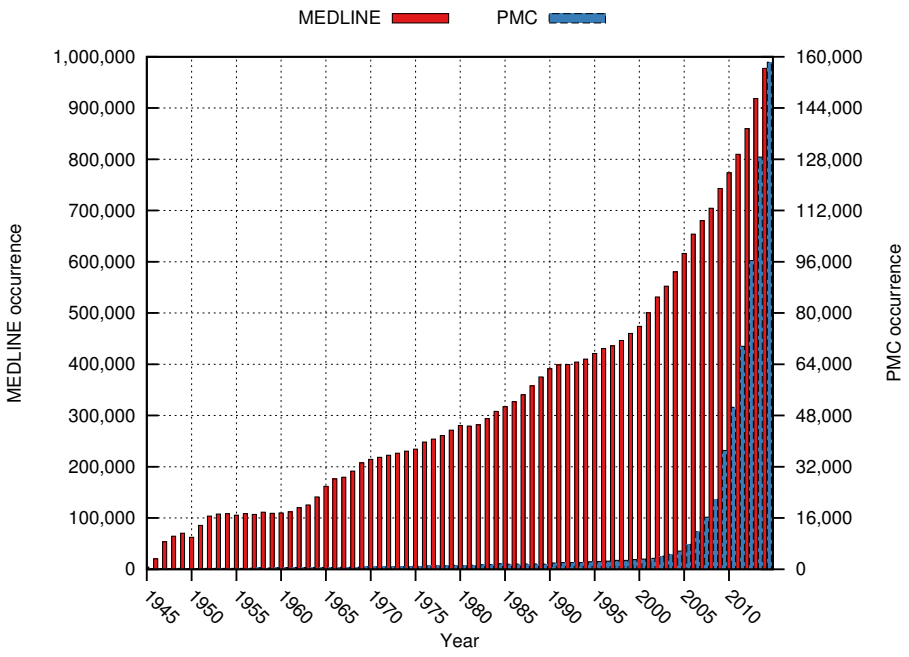


Figure 1.1: Number of biomedical articles published since 1945 until 2014. Results for MEDLINE citations and PMC full-texts use left and right scale respectively.

first. Databases therefore employ specialized personnel (called curators) to transfer relevant information from recent publications to the database. A severe problem for curators is that relevant information is typically not published in one specific journal, but rather spread across many journals. BIND (Bader *et al.*, 2003) curators surveyed a large range of journals and estimated that more than 1,900 interactions are published in almost 80 different journals per month (Alfarano *et al.*, 2005). Data is not only spread in different journals, but also fragmented into different databases. De Las Rivas and Fontanillo (2010) analyzed the overlap of human protein-protein interactions from six different databases and identified only three interactions contained in all resources.

Without any major advances in curation technology, curation times are going to be very high. In fact, Baumgartner *et al.* (2007) estimated a linear or even slower increase of missing information for manual annotation. According to their results, association of all mouse genes with at least one Gene Reference Into Function (GeneRIF[4]) annotation is not going to be complete before 2045. It is worth mentioning, that several manually curated databases (*e.g.,* Kegg or TAIR) have been recently hit by funding cuts, leading to a decrease in available curators and curated entries. Several initiatives tried to reduce the burden from biocurators by motivating authors to transfer their findings into databases (Seringhaus and Gerstein, 2007; Giardine *et al.*, 2011). However, most researcher seem to be rather database consumers and rarely contributors (Mazumder

[4] `http://www.ncbi.nlm.nih.gov/gene/about-generif`

et al., 2010). One reason for this behavior is that scientists gain prestige by publishing results in scientific journals and not by adding new data into databases.

Another issue is that biological knowledge is not static. Therefore, annotated information has to be updated on a regular basis. For instance, early protein sequencing methods often missed the first amino-acid, leading to wrong protein sequences. This problem is often acknowledged in protein-substitutions where the position was often derived on the old protein sequence (Yip *et al.*, 2007). Long-term maintained databases have to be constantly updated and sanity checked. For instance, after 10 years 16 % of all entries in a disease database required some sort of curation (Giuse *et al.*, 1995).

The previous paragraphs illustrated that abundant structured and semi-structured information is rapidly growing. Curators alone cannot keep up with the fast increase of published information and researchers have little incentives to transfer their findings into structured databases. Text mining systems offer a way to handle the emerging volume of semi-structured texts (Zweigenbaum *et al.*, 2007). Increasing availability of computational resources together with constantly improving informational extraction tools enable the application to large text repositories (such as MEDLINE and PMC), to support database curators in their every day work. For instance, the recognition and normalization of named entities can be used to support end-users for document retrieval.

1.2 Goals and Contribution

A large body of publications has been presented for biomedical relationship extraction. Many publications focused on the task of protein-protein interaction extraction. This thesis covers the extraction of binary relationships from biomedical texts. Depending on the experimental setting, we work on the domain of drug-drug interactions or protein-protein interactions. The goal of this thesis is the development of robust relationship extraction methods. Specific contributions to the objective of relationship extractions are as follows:

- We implement a machine learning framework benchmarking different relationship extraction methods. This framework is applied to predict drug-drug interactions on two different domains (MEDLINE articles and cleansed HTML pages). By combining the two corpora we estimate domain specificity of learned classifiers. The performance of individual classifiers is improved by applying ensemble learning techniques.

- We introduce self-training to improve robustness for protein-protein interaction extraction on texts with unknown text characteristics. Performance is evaluated using extrinsic studies, where a relation extraction algorithm is evaluated on a corpus different from the training corpus.

- We discuss the distant supervision paradigm and implement it to create an automatically labeled corpus from unannotated text. We compare two different relationship extraction models on this corpus. First, we learn a statistical model and

introduce a series of preprocessing steps to improve the quality of the automatically labeled corpus. Second, we describe a method to learn graph patterns from this corpus. We present a series of steps to refine the pattern set to improve precision or recall.

1.3 Outline of this Thesis

Chapter 2 provides an introduction into important concepts relevant throughout this work. The main focus of this chapter are text mining approaches, as well as an introduction into machine learning and evaluation concepts. The chapter concludes with a survey of related work.
Chapter 3 presents our approach for drug-drug interaction extraction. We show that the aggregation of individual relationship extraction methods improves overall performance by decreasing the risk of choosing an overfitted classifier.
Chapter 4 discusses the problem of domain dependence, which leads to reduced performance when applying a model on a text corpus with unknown characteristics. Using cross-learning studies (*i.e.,* training on one corpus and testing on a different corpus) we quantify the impact of domain dependence in protein-protein interaction extraction. We propose and evaluate the use of two different self-training procedures to reduce domain dependence.
Chapter 5 introduces the concept of distant supervision to label a large text corpus without manual intervention. Using this corpus we train two different relationship extraction methods and compare them on five common test corpora.
Chapter 6 describes the architecture of our semantic search engine GeneView for biomedical texts. We discuss computational requirements to build this resource from scratch and present some applications utilizing the data extracted by different state-of-the-art components.
Chapter 7 summarizes the main contributions of this thesis and ends with an outlook to future work.

1.4 Own prior Work

Some chapters of this thesis are based on work which has been published previously in peer-reviewed publications.

Chapter 3 extends our contribution for the SemEval 2013 shared task (Thomas *et al.*, 2013b). The contributions of this chapter can be attributed to the authors as follows: Thomas conceived the experiments, converted the different corpora into a common XML format, injected parse-tree information, produced results for the following relationship extraction tools (APG, SL, ST, SST, SpT, and PT), and built different ensembles. Neves produced predictions for Moara and TEES including results for re-labeling. Rocktäschel implemented the relationship extraction method SLW and produced predictions for this method. Leser supervised the work. The manuscript was drafted by all authors.

Chapter 4 contains unpublished work and has been developed in conjunction with Solt and Leser. Thomas conceived, performed, and evaluated the experiments. Solt helped with a MEDLINE wide application of existing NLP components (*i.e.,* Charniak Lease parser). Leser supervised the work.

Chapter 5 presents our results using distant supervision. The first part, focusing on the use of support-vector-machines, has been published in Thomas *et al.* (2011b). The contributions of this chapter can be attributed to the authors as follows: Thomas conceived and performed the experiments. Solt helped analyzing the data. Leser in cooperation with Klinger supervised the work and revised the manuscript. The manuscript was written by all authors.

The second part, utilizing graph patterns has been previously published in Thomas *et al.* (2011c). Pietschmann developed and implemented the ideas to filter and generalize patterns in order to improve precision or recall respectively. Thomas resolved problems with the evaluation strategy, extracted patterns from PMC and MEDLINE, re-performed all experiments, and revised larger parts of the original code. Solt helped with the implementation. Leser in cooperation with Tikk supervised the work and revised the manuscript. The manuscript was written by Thomas, Solt, Tikk, and Leser.

Work on approximate subgraph matching has been performed during a two month research visit at the University of Colorado, Denver in collaboration with Haibin Liu. Liu provided the algorithm for approximate subgraph matching and contributed ideas for pattern ranking.

Chapter 6 presents the architecture of GeneView and some applications, which has been previously published in Thomas *et al.* (2010), Thomas *et al.* (2012a), and Thomas *et al.* (2013a). Thomas implemented the parallelized workflow for all text-mining tools (including text and data storage, named entity recognition, relationship extraction) and performed the large scale evaluations (except for pathway reconstruction). Starlinger implemented the front end of GeneView, including the XML-RPC Lucene interface. Vowinkel provided modifications to the front end and implemented the character-mapping required for the visualization of PMC full text articles. Arzt implemented parsers for PubMed and PubMed Central XML and performed the pathway reconstruction experiments (Subsection 6.4.4). Jacob determined appropriate section weights using the gene2pubmed database. Leser supervised the work. The manuscript was written by Thomas, Starlinger, and Leser.

2 Biomedical Text Mining

In the digital century, text is plentiful available in different resources such as web-pages, digitalized books, news paper articles, or scientific publications. The goal of biomedical text mining is to transform the unstructured written language into computer readable structured data to support life science research. One particular challenge is the high ambiguity of natural language allowing to express a given fact using many different ways. Another type of ambiguity are homonyms, which denote words having identical syntactic base form but different meaning. For instance, the word "bow" may refer to the weapon, the action of somebody bending down, or the front of a ship. A text mining system has to be capable of handling these problems. Such tools can support humans in tedious and time consuming tasks and can reduce manual efforts. For instance, FlyBase curators reported a decrease of annotation time by 20 % when using natural language processing tools for their assistance (Karamanis *et al.*, 2007). Other tasks are targeted retrieval of relevant articles, markup of named entities for subsequent manual curation, automatic pathway-reconstruction, and many more. Some of these tasks will be discussed in more detail in Chapter 6.

This chapter gives an introduction on relevant text mining and machine learning concepts, describes standard evaluation metrics and procedures, and closes with an introduction to related work for relationship extraction.

2.1 Natural Language Processing

Natural language processing (NLP) describes the ability of a computer program to analyze natural language. This thesis covers the analysis of written text, but NLP may also refer to the analysis of spoken text. A large number of specific NLP tasks exists and this section explains frequent steps a typical text mining system has to carry out to analyze written texts. The workflow of a simple information extraction system is shown in Figure 2.1, but a specific implementation might comprise different analysis steps.

2.1.1 Sentence Boundary Detection

Sentences are often used as "informational unit" for the extraction of information from text. This assumption has been tested in the context of relationship extraction for different domains. For instance, Swampillai and Stevenson (2010) reported for the ACE03 news corpus that 90.6 % of all relations are mentioned within the same sentence. Similar results are reported by Björne *et al.* (2009) for the BioNLP'09 corpus where approximately 95 % of all biomedical events are stated within the same sentence. Although this assumption is not always correct, it is a helpful heuristic exploited in several text

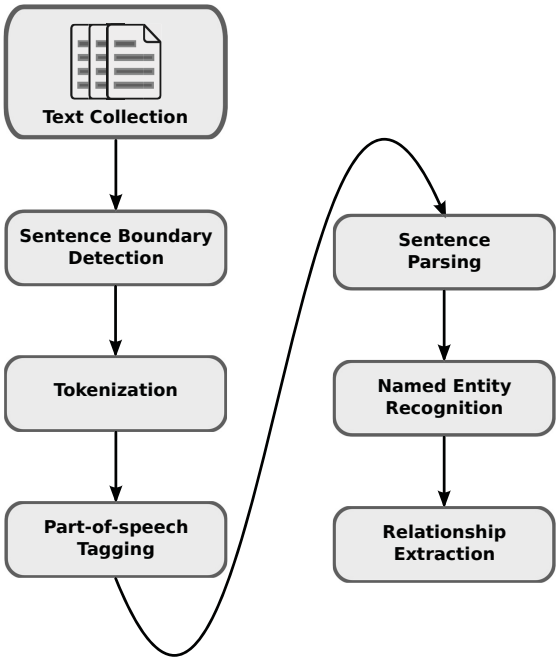


Figure 2.1: Example for a text mining workflow.

mining approaches. Furthermore, several NLP tasks, such as part-of-speech taggers and syntactic parsers, require properly recognized sentence boundaries.

Sentence boundaries are not explicitly marked and need to be detected by a separate method. This allows to split articles into smaller fragments, which can then be handled by subsequent NLP components. Sentence boundary detection is a non trivial task, as punctuation marks (., !, ?) are not always an explicit indicator for sentence boundaries. Examples include abbreviations such as Mr. or St., company names such as Yahoo!, or embedded questions. It is worth mentioning that the heuristic using only punctuation marks to determine sentence boundaries still achieves a 90 % accuracy in the "Brown University Standard Corpus of Present-Day American English" (Riley, 1989). The very same heuristic achieves only an accuracy of 53 % on the "Wall Street Journal" corpus (Stamatatos *et al.*, 1999). Recent methods, based on Conditional Random Fields (Lafferty *et al.*, 2001), solve the problem of sentence boundary detection with approximately 99.6 % accuracy for biomedical text (Tomanek *et al.*, 2007). According to their analysis, domain specificity is not a problem as sentence boundaries are vastly uncontroversial.

2.1.2 Tokenization

Tokenization refers to the segmentation of sentences into atomic text units. Unfortunately, the definition of "atomic" units is highly language and domain specific. For instance, some East Asian languages, such as Chinese or Japanese, are written without

separating word spaces (Nakagawa, 2004). This leads to tokenization problems that are different from the modern western languages. In English problematic cases are for instance, monetary amounts such as $9.99, hyphenations such as e-mail, apostrophes such as isn't, phone numbers such as (00)43 123-456789, or proper nouns such as C#.

Domain specific problems can be observed in the biomedical domain as well, where special characters may occur as part of a biomedical name. An example are parenthesis in words such as "CD34(+)", where the substring '(+)' indicates presence of an antigen. Other examples are hyphens occurring as part of mutations (*e.g.,* Cys32-Gly), as part of a gene name (*e.g.,* Rev-Erbα), or as part of disease names (*e.g.,* Glio-Sarcoma). Hyphenations can also be used to indicate fusion genes (*e.g.,* the oncogene "BCR-ABL"). In all these cases, a tokenizer should recognize the semantic unit and not split the hyphenation.

Tokenization algorithms usually are either rule-based (as implemented in Lucene) or machine learning based. Tomanek *et al.* (2007) achieve an accuracy of approximately 96 % for token boundary detection using CRFs on the PennBioIE corpus (Bies *et al.*, 2005).

2.1.3 Part-of-Speech Tagging

Another common pre-processing step is the prediction of part-of-speech (POS) tags for each token. The English language generally contains the following eight lexical categories: noun, pronoun, adjective, verb, adverb, preposition, conjunction, and interjection. Tag sets, such as the Penn Treebank (PTB) (Marcus *et al.*, 1993) tag set, often use more fine-grained subcategories. For instance, nouns are distinguished into singular noun, plural noun, singular proper noun, and plural proper noun. The PTB tagset distinguishes 36 POS tags and 12 other tags (*e.g.,* for punctuations and brackets). State-of-the-art approaches achieve a POS-tagging accuracy of $> 97\,\%$ for the Penn corpus (Toutanova *et al.*, 2003) and similar performance can be achieved for biomedical corpora (Smith *et al.*, 2004). It is worth pointing out that a simple heuristic presented by Charniak *et al.* (1993) achieved 90.25 % accuracy by assigning the most common tag to each known token and the tag proper noun to unknown words.

Part-of-speech tags can be used to reduce word ambiguity. This allows to differentiate ambiguous words *e.g.,* the noun 'book' from the verb 'book'. This information is often used in relationship extraction to increase pattern specificity (Hakenberg *et al.*, 2010; Liu *et al.*, 2010b).

2.1.4 Sentence Parsing

Parsing refers to the syntactic analysis of a sentence with respect to some formal grammar. Parsing allows to group words and syntactically relate them to each other, generally resulting in a syntactic parse tree. The goal of parsing is to select the most probable structural parse tree for a given sentence. This is often accomplished by generating a large number of valid parse trees (wrt the chosen grammar), computing their likelihood, and selecting the most probable parse. This can be difficult for syntactically ambigu-

ous sentences such as "The man saw the moon with a telescope.", where two syntactic interpretations are possible. The most likely interpretation is that the man looked at the moon using his telescope. However, this sentence could also be understood as the telescope is located on the moon. While the second interpretation seems improbable to a human reader it is syntactic and semantically valid.

Sentence parsing is widely acknowledged as an important step in biomedical relationship extraction. For instance, 20 out 24 participating teams used parsing in the BioNLP'09 shared task (Kim *et al.*, 2009). Two alternative parse structures are commonly used and will be explained in more details in the following subsections.

Constituent tree parsing

Constituency syntax has long dominated theoretical and computational linguistic research since the seminal work of Chomsky (1957). In constituency theory, words or group of words are hierarchically organized as constituents. Therefore the number of nodes in the constituency tree equals the number of constituents. The most commonly used representation for constituent tree parsing is the Penn Treebank scheme (Marcus *et al.*, 1993). Two possible constituent parse trees are shown in Figure 2.2, for the semantic ambiguous sentence "The man saw the moon with a telescope.". Both parses are identical except for the attachment of the prepositional phrase "with a telescope". In Figure 2.2(a) the man looked at the moon using his telescope, whereas in Figure 2.2(b) the telescope is placed on the moon.

Dependency tree parsing

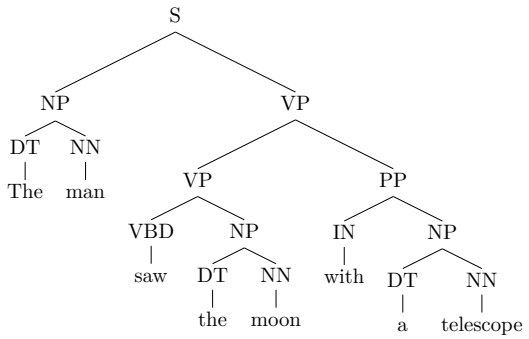
In dependency grammar syntactic relationship is represented as typed directed binary relation from the govern (head) to its dependent (child). Such a dependency link represents how two words relate to each other. All nodes, except the root node, depend on one dependent; but two or more words can have the same govern by separate typed dependencies. According to Tesnière, the inventor of modern dependency theory, the root node is generally the main verb of a sentence and the dependency type (edge label) represents the grammatical relation between two words (Tesnière, 1959). Several different dependency schemes exist, where the Stanford representation is probably most often used (De Marneffe and Manning, 2008). Dependency trees for our semantically ambiguous working example are shown in Figure 2.2(c) and 2.2(d).

Differences

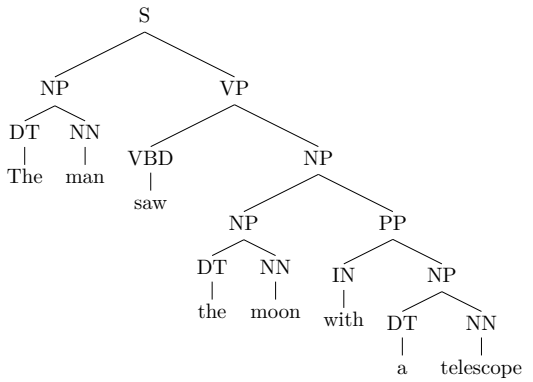
Several differences between constituency and dependency syntax exist. In constituent trees, words appear only as leaves and internal nodes are always non-terminal symbols (*e.g.,* noun phrase, verb phrase, prepositional phrase, ...). The result of constituency parsing is always a rooted tree. Dependency grammar incorporates only tokens from the original sentence and hence neglects the concept of non-terminal nodes (as used in constituency grammar). Depending on the selected dependency scheme, the dependency

graph may contain cycles but no self-dependencies. Another difference is that constituency trees follow the linear text order, whereas dependency grammars, if following the original proposal of Tesnière (1959), allow non-linear word order.

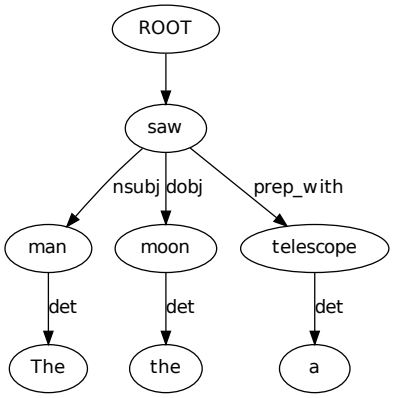
Several researchers noted an increasing interest in dependency based representations (Nivre, 2005). Similarly, a strong favor towards the dependency grammar can be observed for biomedical relationship extraction. For instance, 10 of 12 teams in the BioNLP'11 shared task used the Stanford dependency scheme for building their system (Kim *et al.*, 2011a). This tendency is also observed in protein-protein interaction extraction, where a majority of groups uses dependency grammar (see Table 2.3). A frequently found argument is that dependency parses are easier to interpret because of the attachment of semantically related words.



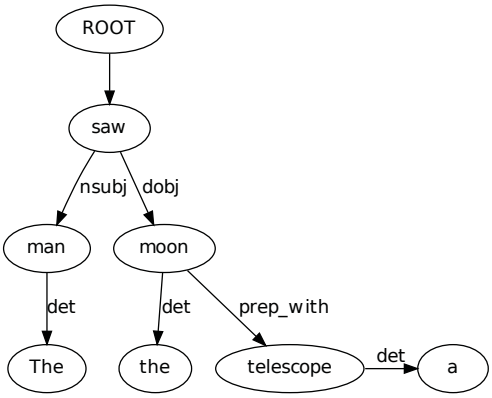
(a) Constituent parse tree where the man looks at the moon using a telescope.



(b) Constituent parse tree where the telescope is on the moon.



(c) Dependency parse tree where the man looks at the moon using a telescope.



(d) Dependency parse tree where the telescope is on the moon.

Figure 2.2: Constituent and dependency parses for the sentence "The man saw the moon with a telescope".

2.2 Machine Learning

The goal of machine learning is to learn a statistical model capable of generalizing from training examples. In this section we will focus on binary classification, where the goal is to learn a model capable of assigning a class label $y \in \{0, 1\}$ to every provided instance. Each instance can be represented by a m dimensional feature vector $\mathbf{x} = (x_1, \ldots, x_m)$, where each feature x_i represent individual observations. To generate this representation, instances need to be transformed from the input space (text) into the m dimensional feature space representation using a mapping function ϕ. For supervised learning, the learning algorithm uses n labeled instances of the form $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ where y_i is the associated class of feature vector $\mathbf{x_i}$ for each instance i. The learning algorithm then learns a statistical model on the training instances, which can be used to predict class labels (y) on an instance using the same m dimensional feature representation.

For example, Spam detection can be formulated as classification problem, where every instance (mail) is classified as Spam or not Spam. Before training a classifier, all mails need to be transformed into the feature space. For example, features can indicate the presence or absence of specific tokens or the number of tokens per mail.

A multitude of methods have been proposed for classification. Some of the best known methods are K-nearest neighbors, Naïve Bayes, or decision trees. Support vector machines (SVM), another machine learning algorithm, is, due to promising empirical performance, one of the most widely used classifiers in bioinformatics (Ben-Hur *et al.*, 2008; Irsoy *et al.*, 2012) and will be explained in the following subsection.

2.2.1 Support Vector Machine

This subsection provides an introduction on SVM and is loosely based on the excellent presentations of Cristianini and Shawe-Taylor (2003) and Ben-Hur *et al.* (2008). SVM is a linear classifier (Boser *et al.*, 1992), which is intuitively described as "find a hyperplane that separates positive from negative instances best in the given feature space". The *best* hyperplane is defined as the one maximizing the margin between positively and negatively labeled instances. This intuition is exemplified in Figure 2.3, showing the hyperplane maximally separating positive and negative instances.

To explain the concept of SVM we first define a linear discriminant function $f(x)$ as:

$$f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b \tag{2.1}$$

The function $f(\mathbf{x})$ assigns a score to the unlabeled instance $\mathbf{x}$, given the weight vector $\mathbf{w}$ and the bias scalar b. $sign(\langle \mathbf{w}, \mathbf{x} \rangle + b)$ predicts the class y for instance $\mathbf{x}$. This function allows to separate the feature space into two separate parts with positive instances above and negative instances bellow the hyperplane. The function described in Formula 2.1 defines an arbitrary linear classifier without considering the maximum margin principle. To find the hyperplane maximizing the margin ($\frac{1}{\|\mathbf{w}\|}$) between the positive and negative

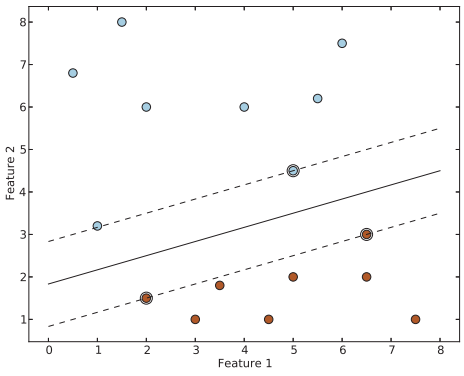


Figure 2.3: A linear classifier separating two classes by the maximal margin principle. Blue and red dots represent training instances from two different classes. The solid line represents the learned decision boundary. The area between the two dashed lines indicates the maximum margin area. Framed data points are called support vectors. These data points are defined as closest to the hyperplane with a distance of 1. Figure drawn using the machine learning tool Scikit-learn (Pedregosa *et al.*, 2011).

instances, we solve the following quadratic optimization problem:

$$\begin{aligned} &\arg\min_{(\mathbf{w},b)} \frac{1}{2}\|\mathbf{w}\|^2 \\ &\text{subject to: } \forall_{i=1}^{n} : y_i \cdot (\mathbf{w} \cdot \mathbf{x_i} - b) \geq 1 \end{aligned} \tag{2.2}$$

Soft margin

So far we formulated the so called hard margin SVM, which requires linearly separable data to work properly. In practice, data sets are not always linearly separable and exact separation can also lead to poor generalization performance. A solution for non-linearly separable data can be found by the introduction of so called slack variables (Vapnik, 1995). Slack variables (ξ_i) are defined as zero for data points located on or outside the margin. Data points with $0 < \xi_i \leq 1$ are correctly classified, but lie within the margin and elements with $\xi_i > 1$ are misclassified. Expanding the inequality constraint in Equation 2.2 with slack variables leads to the following constraint:

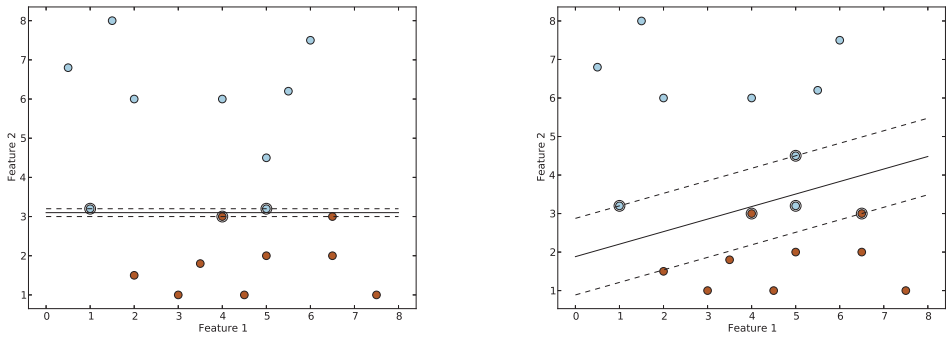
$$y_i(\mathbf{w} \cdot \mathbf{x_i} - b) \geq 1 - \xi_i, \quad \xi_i \geq 0 \tag{2.3}$$

In order to penalize classification errors, the cost-parameter $C > 0$ is added. Large C penalize misclassified instances, whereas small values for C tolerate misclassification.

Altogether, this leads to the formulation of the soft-margin SVM:

$$\begin{aligned} &\arg\min_{\mathbf{w},b,\xi} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i \\ \text{subject to: } &\forall_{i=1}^{n} : y_i(\mathbf{w}\cdot\mathbf{x_i} - b) \geq 1 - \xi_i \\ &\forall_{i=1}^{n} : \xi_i \geq 0 \end{aligned} \tag{2.4}$$

Setting $C = \infty$ we obtain the regular hard margin SVM (Formula 2.2). The impact of high and low C values is shown in Figure 2.4. High values of C, as shown in Figure 2.4(a), imitate the behavior of the hard margin SVM by punishing misclassification of individual instances. Lower values of C allow a larger margin by increasing number of misclassifications on the training set. Without more information on the underlying sample distribution, it remains unclear which separating hyperplane provides a better generalization. However, soft-margin SVM provides a way to reduce the impact of outliers.



(a) $C = 100$ (b) Decision boundary with $C = 1$

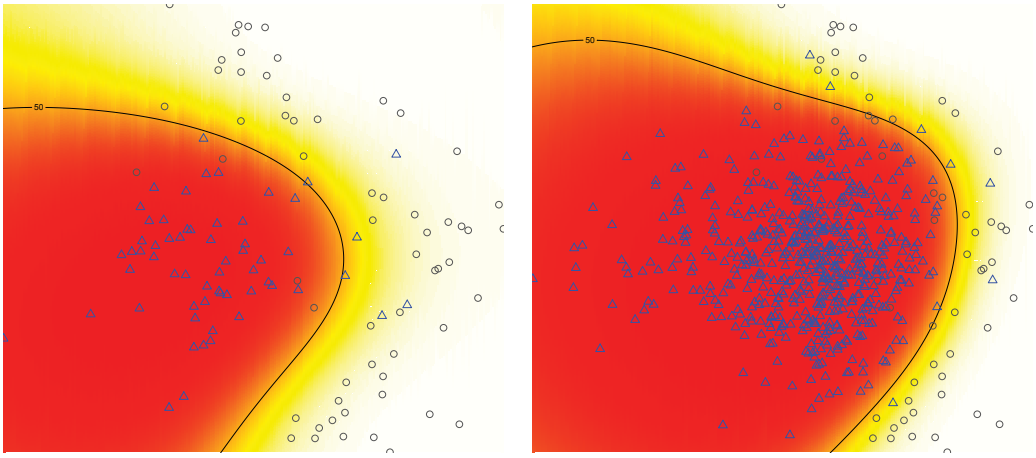
Figure 2.4: Impact of soft-margin constants C on the decision boundary. In the left example misclassification is penalized much harder than in the right example and therefore returns a hyperplane with no misclassification but comparably smaller margin.

Imbalance in class distribution

Datasets with unbalanced class distribution (*e.g.,* 20 times more negative instances than positive) pose a problem to many machine learning classifiers as most methods tend to predict the majority class (Weiss and Provost, 2001; Van Hulse *et al.*, 2007). However, in many classification settings we are more interested in finding the minority class than the majority class. For instance, we are more interested in reliably finding sentences describing a specific relationship than sentences describing no relationship. Most likely,

the majority of sentences describes no relevant relationship. The impact of different class distribution for SVM is shown in Figure 2.5, where we generated two data sets using the same probability distribution. The first data set, shown in Figure 2.5(a), has an identical amount of instances for both classes, whereas in Figure 2.5(b), we oversampled one class 10 times. For both data sets we learned a SVM using default parameters and a Pearson universal kernel (Üstün *et al.*, 2006). It can be seen that the learned hyperplane differs for the two data sets. The SVM learned on the dataset with highly imbalanced class distribution features a higher probability of classifying unlabeled instances into the majority class.

In SVM this problem is usually solved by applying different soft-margin costs (C_{+1} and C_{-1}) (Veropoulos *et al.*, 1999). For instance, misclassification costs can be set 20 times higher for negative than for positive instances.



(a) Learned SVM hyperplane for identical class distributions.

(b) Learned SVM hyperplane with 10 time oversampling of one class.

Figure 2.5: Learned decision boundary for two datasets. Data points have been sampled from the same probability function, but the two different datasets have different class ratios.

2.2.2 Kernels

Linear separation sometimes lacks the expressive power to deal with real world applications. A first approach to non linear classification is to project all instances using a non linear mapping function into a new feature representation and learn an arbitrary linear classifier (*e.g.,* a SVM). A simple example for a non-linear mapping function is the transformation from a two dimensional feature space to a three dimensional space using the mapping function "$\phi(\mathbf{x}) = (x_1^2, x_2^2, x_1 x_2)$". We define the quadratic mapping function for a n dimensional feature space as follows:

$$\phi(\mathbf{x}) = \{x_i x_j | i, j \in \{1, \ldots, n\} \wedge i \leq j\} \tag{2.5}$$

This mapping function leads to a combinatorial explosion for high dimensional feature spaces prohibiting the explicit representation of the transformed feature space. However, a particularity of SVM (and some other algorithms) is that the explicit feature space transformation is not needed. To understand this property we need to make a small detour. Vapnik (1995) showed that the original primal minimization problem (2.4) can be reformulated into its dual form, where α_i are Lagrangian multipliers:

$$\begin{aligned} &\arg\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle \\ &\text{subject to: } \sum_{i=1}^{n} \alpha_i y_i = 0 \ \wedge \ \forall_{i=1}^{n} : 0 \leq \alpha_i \leq C \end{aligned} \tag{2.6}$$

The dual representation has some advantages over the original primal formulation. It can be seen that this dual representation depends on the data only in terms of a dot product. This dual representation allows to replace the dot product by a kernel function. A kernel function between two instances $\mathbf{x}$ and $\mathbf{y}$ is defined as:

$$K(\mathbf{x}, \mathbf{y}) = \langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle \tag{2.7}$$

Using the quadratic mapping function $\phi(x)$ (see Formula 2.5) allows us to rewrite the kernel function as:

$$K(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} \sum_{j=1}^{n} x_i x_j y_i y_j \tag{2.8}$$

This kernel function (Formula 2.8) calculates the distance between two vectors, without the explicit construction of the transformed feature space. It can be seen that the kernel function has constant space requirement. Whereas the explicit formulation has quadratic space requirement, which is often not feasible.

Convolution kernels

Convolution kernels are a specific instance of kernel functions, "which involve a recursive calculation over the parts of a discrete structure" (Collins and Duffy, 2001). In relationship extraction, convolution kernels are frequently used to define a similarity measure between two syntactic parses. These functions usually count the number of shared subtree structures between two trees. A (parse) tree T can be represented as a vector of composing subtrees (see for example Figure 2.6(b)) using the mapping function:

$$\phi(T) = (subtree_1, subtree_2, \ldots, subtree_n) \tag{2.9}$$

Each feature ($subtree_i$) denotes the observation of a specific subtree. Prominent examples based on subtree similarity are subtree (ST) (Vishwanathan and Smola, 2002), subset tree (SST) (Collins and Duffy, 2001), and partial tree (PT) (Moschitti, 2006). These differ only in the definition of subtrees: ST generates subtrees considering all

descendants of any node. SST relaxes this constraint and allows to add either none or all children for a given node in the tree. The PT representation is the most tolerant and allows virtually any subtree structure. The subtrees sets are therefore subsets of each other: ST $\subset$ SST $\subset$ PT. Different subtree representations for the sentence "Bill bought a book" are shown in Figure 2.6. The similarity between two trees is derived as follows: Let N_1, N_2 be the set of nodes in the respective trees T_1 and T_2. The kernel function (2.10) counts the number of identical subtrees rooted at $n1$ and $n2$.

$$K(T_1, T_2) = \langle \phi(T_1), \phi(T_2) \rangle = \sum_{n_1 \in N_1} \sum_{n_2 \in N_2} \Delta(n_1, n_2) \tag{2.10}$$

2.3 Evaluation

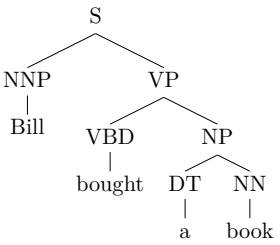
In most real world applications the learning algorithm deals with a large feature space and comparably little training instances. In other words, the feature space is sparsely occupied by training examples. For instance, given a rather small space of 100 binary features requires 2^{100} instances to fully occupy the feature space. Therefore, generalization is a critical feature of a successful machine learning algorithm. A classifier simply memorizing the training examples achieves perfect results on training data, but usually has little generalization abilities on unseen instances. A classifier achieving excellent results on training data but mediocre results on test-data is called over-fitted. Over-fitting can be avoided by artificially separating training and test data. The two most commonly known approaches are bootstrapping and cross-validation, which we will cover later in some detail.

The performance of a method is measured using a gold standard. Manually annotated data is usually used to estimate performance, but also other sources (*e.g.,* knowledge bases) can be used. Measures reported in this thesis are based on the so called confusion matrix exemplified in Table 2.1. The individual entries represent observed frequencies of instances being correctly classified as positive (TP), instances wrongly identified as positive (FP), instances wrongly identified as negatives (FP), and instances correctly predicted as negatives (TN). A perfect classifier would only fill the diagonal (TP and TN), yielding 100 % correct predictions. The observations are used to calculate the metrics *precision* and *recall* which are defined as follows:

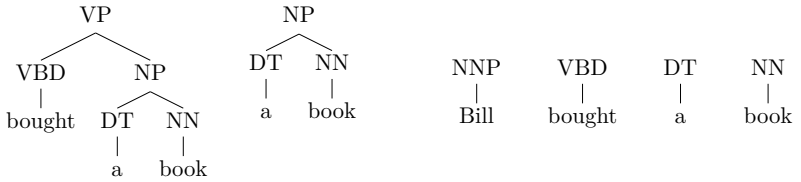
$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2.11}$$

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2.12}$$

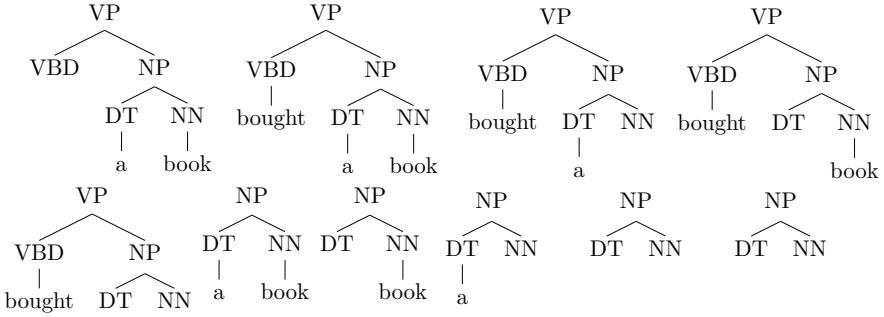
Often a tradeoff between recall and precision has to be found, because optimizing a system to higher recall usually lowers precision and vice versa. For instance, an information retrieval system returning all contained documents for an arbitrary query trivially has 100 % recall but should display low precision. Therefore, the goal is to find an agreement between precision and recall. One prominent way is the F_β-measure (see



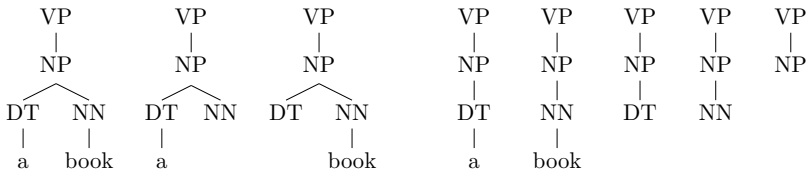
(a) Original parse tree



(b) Subtrees using ST representation (Vishwanathan and Smola, 2002).



(c) Subset of subtrees using SST representation (Collins and Duffy, 2001).



(d) Subset of subtrees using PT representation (Moschitti, 2006).

Figure 2.6: Different subtree representations for the constituency parse "Bill bought a book".

		Real	
		positive	negative
Prediction	positive	TP	FP
	negative	FN	TN

Table 2.1: Example for a confusion matrix with two classes (positive and negative).

Formula 2.13), which is based on to the effectiveness measure (E-measure) introduced by Van Rijsbergen (1979, p. 174). The F_β-measure is the weighted harmonic mean of precision and recall. The factor β allows to emphasize on precision or recall, but in most settings $\beta = 1$ is used. Precision as well as recall neglect true negative predictions as these numbers are usually very large and would put too much emphasis on true negative predictions. Accuracy, as defined in Formula 2.14, incorporates all four characteristics and is generally not recommended in a setting with highly imbalanced class distribution.

$$F_\beta = \frac{(1+\beta^2) \cdot \text{precision} \cdot \text{recall}}{\beta^2 \cdot \text{precision} + \text{recall}} \tag{2.13}$$

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{2.14}$$

The harmonic mean will always decrease when precision and recall are subject to a mean preserving spread (Mitchell, 2004). This means that the F_1 measure penalizes diverging differences between recall and precision, although the arithmetic mean remains constant. A visualization of this property is shown in Figure 2.7 where F_1 values are shown as a function of precision and recall.

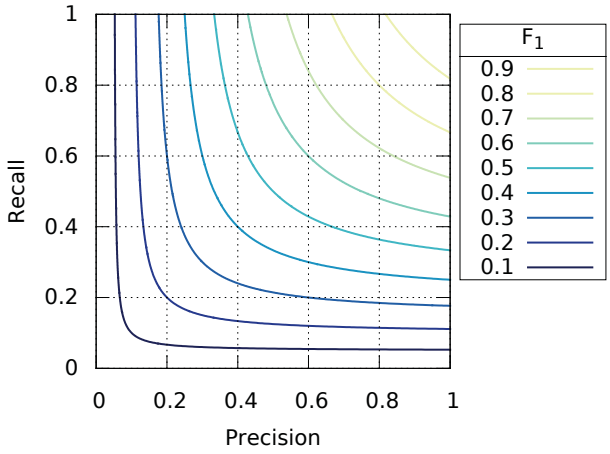


Figure 2.7: Relationship between precision and recall for predefined F_1 values.

Another evaluation metric is the so called receiver operating characteristic (ROC), or ROC curve (Egan, 1975). Several binary classifiers produce not a discrete output (*i.e.,* class label), but rather a continuous value which can be used as a confidence measure. For instance, for SVMs the distance to the hyperplane can be used, where large absolute values indicate higher certainty and values close to the separating hyperplane (close to zero) indicate low certainty. ROC curves visualize performance of binary classifiers over varying discrimination thresholds, where the x-axis represents the false positive rate (FPR) and the y-axis the true positive rate (TPR). For a definition of TPR and FPR see Formula 2.15. A ROC curve is shown in Figure 2.8. In the plot the point (0,0) corresponds to all instances classified as negative, whereas the point (1,1) represents

all instances classified as positive. Perfect classification is achieved at point (0,1). An import property of ROC curves is that they are insensitive to class distribution. The curve is often aggregated to a single value; the so called area under the ROC curve (AUC). AUC corresponds to the probability that a random positive instance achieves a higher confidence score than a random negative instance (Hand and Till, 2001). An important disadvantage of AUC analysis is that even a well-fitted model (high AUC) might achieve moderate discrimination performance. For example, let us assume that a classifier assigns a score of 0.99 to all positive instances and a score of 0.98 to all negative instances. This model achieves perfect discrimination (*i.e.,* AUC = 1), but finding the sweet spot for a discriminative threshold is rather difficult.

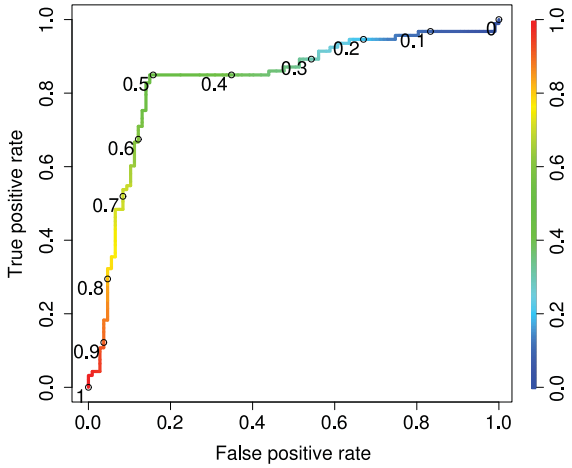


Figure 2.8: ROC curve for a Naïve Bayes classifier on an arbitrary dataset. Color indicates varying classifier thresholds. Individual points mark specific thresholds. Visualization performed using ROCR (Sing *et al.*, 2005).

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \qquad \mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{TN} + \mathrm{FP}} \tag{2.15}$$

2.3.1 Model Validation

The previous section introduced precision, recall, F_1, and AUC as evaluation measures. These are calculated on unseen test-data in order to get realistic estimates. The generalization of a model is often assessed by k-fold cross-validation (Geisser, 1975) or bootstrapping (Efron, 1979). In k-fold cross-validation all available data $\mathcal{D}$ is partitioned into k disjoint parts $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_k$ of similar size. A classifier is trained on the union of $k-1$ subsets and evaluated on the remaining and unseen test set. This strategy is repeated $i \in \{1, 2, \ldots, k\}$ times, where the classifier is trained on $\mathcal{D} \setminus \mathcal{D}_i$ and evaluated on $\mathcal{D}_i$. The most extensive cross-validation procedure is leave-one-out where the dataset is par-

titioned into $k = |\mathcal{D}|$ parts. The advantage of leave-one-out cross-validation is the high amount of available training data, but the approach can be computationally infeasible for larger data-sets. In practice, tenfold cross-validation is often recommended (Kohavi, 1995).

Bootstrapping samples $|\mathcal{D}|$ instances from the original dataset using random sampling with replacement. A classifier is trained on the drawn samples and evaluated on the remaining unseen samples. To get sensible estimates this procedure is repeated k times. With increasing data set size n the likelihood of selecting an instance i for training asymptotically approaches $P(i|n) \sim 1 - e^{-1}$. Kohavi (1995) argue that cross-validation and bootstrap are helpful performance estimation techniques which can lead to wrong estimates on some data sets. For PPI extraction, document-wise 10-fold cross-validation is by far the most frequently used estimation technique.

Cross-validation or bootstrapping work well when applied with caution, but overestimate performance when repetitively applied on the same data set (Ng, 1997; Salzberg, 1997). Examples for repetitive performance estimations are exhaustive parameter space exploration, model comparison, or feature selection. In these cases, the peak in performance is likely due to over-fitting to the training data and performance will be significantly lower on unseen data sets. This is especially an issue for sophisticated machine learning methods with many parameters where many different parameter combinations are possible. The problem is related to the multiple testing problem in statistics (Ng, 1997).

2.4 Information Extraction

Information extraction deals with the extraction of previously defined facts from unstructured documents. Two popular subtasks of information extraction are named entity recognition and relationship extraction and will be explained in more detail in this section.

2.4.1 Named Entity Recognition

The goal of named entity recognition (NER) is to identify entities of a previously defined type. Examples are corporations, locations, or person names. Examples for named entities in the biomedical concepts are gene/protein, mutation, or disease names. Unambiguous association of a named entity to a unique canonical form or database identifier is termed named entity normalization.

Several properties of name usages, such as term ambiguity or the partial use of existing nomenclature make NER a difficult task. For instance, several gene names are derived from the phenotype when the gene is absent or depleted. These gene name can overlap with common English words as for the fruit fly gene breathless (FBgn0005592). Additional ambiguous gene names are blood, disco, red, or can (Proux *et al.*, 1998).

Another problem is that researchers tend to neglect nomenclatures and instead prefer previously established synonyms (Tamames and Valencia, 2006). For instance, the official gene symbol XCL1 (Entrez 6375) can be found 311 times in all of MEDLINE,

but the synonym ATAC is mentioned 417 times and the full gene name lymphotactin occurs 255 times. According to the Entrez Gene database (Maglott *et al.*, 2011), the human gene with most synonyms is, with 31 different entries, OR4H6P (Entrez: 26322). The gene with most known synonyms is the drosophila gene tws (Entrez: 47877) with 89 entries. The distribution of synonyms for all human genes has been extracted from Entrez Gene and is shown in Figure 2.9.

Gene names follow no regular structure but can appear as anything from a three letter acronym to a multi-token complex name. These problems exist for other entity types, such as diseases (whose names often contain ordinary persons' names, like "Wilsons disease"), or medical symptoms (whose names can be used in many different contexts not related to diseases, *e.g.,* "shiver" or "cold"). Similar to gene name nomenclature, the mutation nomenclature is not fully adapted by researchers. This example will be discussed in more detail in Subsection 6.4.2.

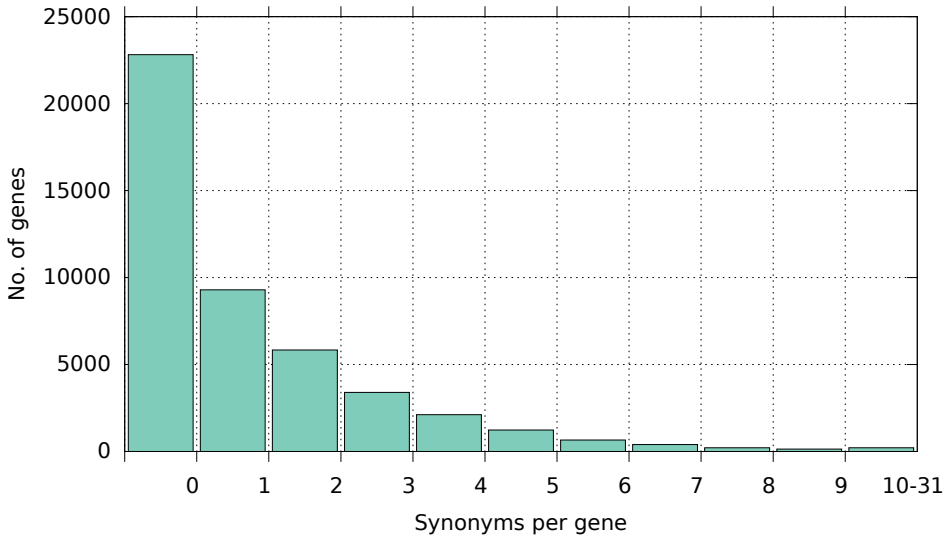


Figure 2.9: Histogram of synonyms for all human genes according to Entrez gene.

A related problem is the introduction of morphological variations. For instance, the human gene BRCA1 is also referred to as BRCA-1, Brca-1, BRCA-I, and many more. Another problem, often mentioned in the context of NER is the uncertainty of exact text boundaries (Wang, 2010). For instance, some annotators annotate species mentions co-located with the protein name (*e.g.,* human hemoglobin), whereas other people only consider "haemoglobin" as the gene/protein mention.

NER should also be able to handle spelling errors, such as "colorecal cancer" (PMID: 16422107), "colorecatal cancer" (PMID: 22202261), or erroneous hyphenation such as "colorec-tal cancer" (PMID: 19663088). Gene names can also be accidentally modified by the activated auto conversion function in word processing tools such as Microsoft Excel (Zeeberg *et al.*, 2004).

In contrast to other domains, where NER is considered as an essentially solved problem (Balke, 2012), biomedical NER remains far from being solved in a satisfying manner. For example, for the recognition of person names, organizations, and geographic locations the best performing team achieved a F_1 of 96 % during the Message Understanding Conference-6 (Grishman and Sundheim, 1996). In contrast performance for gene, chemical, and disease named entities have been estimated at about 61 %, 74 %, and 51 % F_1 respectively during the BioCreative IV-CTD shared task (Wiegers *et al.*, 2014).

2.4.2 Relationship Extraction

The goal of relationship extraction is the detection of relations between named entities. This task gained much attention within the last years and a large set of publications dealing with relationship extraction appeared. In this thesis, we will focus on binary undirected relationship extraction. This annotation scheme has been identified as the greatest common factor for protein-protein interaction corpora (Pyysalo *et al.*, 2008a).

The scientific community often distinguishes three different approaches of relationship extraction, which are not mutually exclusive. The three general approaches are described in the following part of this section. Related work concerning protein-protein interaction will be described in more detail at the end of this chapter and approaches for drug-drug interaction will be explained in Chapter 3.

Co-occurrence

Early approaches used the concept of co-occurrence to detect relations between named entities. The working hypothesis of this approach is that entities mentioned in the same textual context can be expected to share a semantic context. Textual context types are for instance document, paragraph, sentence, or phrase (Ding *et al.*, 2002). Co-occurrence based approaches achieve very high recall and low precision as they predict a relationship for every entity pair in a given context. Depending on the frequency of positive instances, precision for PPI extraction ranges from 17 % to 50 % (Pyysalo *et al.*, 2008a). Co-occurrence is most often used as a baseline to evaluate relation extraction approaches. Precision of co-occurrence can be substantially improved by requiring the mention of an interaction word (Kabiljo *et al.*, 2009) or consideration of other heuristics such as sentence length or the distance between two entities.

An advantage of co-occurrence based approaches is that they require no manually annotated training data and are therefore easy to adapt to novel domains. Another advantage is that they require no sophisticated NLP analysis and are thus often used in large scale applications where run-time is important. The application on a large text repository gives co-occurrence additional strength as frequently found interactions are more likely correct. Some of these frequency based approaches use statistical information measures such as χ^2, mutual information content, or log-likelihood ratio to find significantly overrepresented co-occurrences (Bunescu and Mooney, 2005b; Rebholz-Schuhmann *et al.*, 2007; Hur *et al.*, 2009; Fleuren *et al.*, 2011). Wright *et al.* (2010) showed that statistically motivated approaches often outperform purely frequency based

co-occurrence.

Pattern based

The second type of approaches use a previously defined set of linguistic patterns to extract relationships. Early approaches in the biomedical domain relied on simple patterns in form of "EntityA relation EntityB" (Blaschke *et al.*, 1999). For this work Blaschke *et al.* used a predefined set of 14 verbs (*e.g.,* associated with, bind, suppress, ...) and possible inflections. For instance, the regular expression `regulat(ions?|(e[esd]?))`matches different word inflections of the word regulate. Similar patterns are used by Ono *et al.* (2001) but they define also rules to handle complex sentence structure and negations. Baumgartner *et al.* (2008) manually defined 67 rules[1] using a regular grammar based on words, POS-tags, phrase types, and ontology concepts. Other approaches defined patterns on the dependency graph (Ding and Berleant, 2003; Rinaldi *et al.*, 2006; Fundel *et al.*, 2007).

Originally these pattern based approaches were based on manually defined rules, but also approaches which automatically learn patterns are proposed. Caporaso *et al.* (2007b) explain a strategy to semi-automatically learn surface patterns for the recognition of mutation mentions. Mutations consist of three mandatory arguments (wildtype, location, and surrogate). Therefore, this task can be defined as three-ary relationship extraction problem. Potential patterns are automatically derived from MEDLINE, by searching sentences containing all three arguments. Recognized arguments are replaced by argument specific place holders (*e.g.,* lysine becomes `aminoacid`) to increase generalizability. Patterns are then generated by extracting the shortest span (on the surface level) between all arguments and the words between them. Automatically generated patterns are ranked by frequency and are manually annotated for correctness. The same strategy has been exploited to learn drug-disease relationship patterns (Xu and Wang, 2013) and histone modification patterns (Thomas and Leser, 2013). In all three domains the strategy achieves excellent precision ($> 90\,\%$) on manually annotated corpora. Recall levels at approximately $80\,\%$ for all three domains.

Machine learning

Several systems for information extraction (NER and RE) make use of statistical classifiers learned on manually labeled corpora. Relationship extraction using machine learning is often cast into a binary classification problem. For each sentence with n entities, all $\binom{n}{2}$ possible undirected entity pairs are constructed. The task of the learned classifier is to decide if a specific entity pair interacts or not. The foundations of machine learning are covered in Section 2.2 and a more detailed comparison of supervised PPI extraction methods will be covered in Section 2.5.1.

[1] `http://sourceforge.net/projects/opendmap/files/supplementalPatterns/supplementalPatterns-1.0/`

Summary

Within the NLP domain there is a constant discussion between researchers favoring machine learning or rule-based approaches. Rule-based approaches often achieve excellent precision, but suffer in recall. A frequent argument against pattern based approaches is the high requirement in time and skills to build patterns. For instance, the adoption of an existing rule-based system to a Message Understanding Conference task has been estimated with approximately 1,500 working hours by Lehnert *et al.* (1992). A frequently mentioned advantage of machine learning methods is that adaptation to a new domain is fairly simple. It only requires annotations for the new target domain and after a learning phase the model can be applied on the new domain. However, this is not always the case as the new domain could have distinctive properties which are not covered by the existing system. In these cases it is necessary to modify the machine learning system to cover these distinctive properties.

Recent evaluations indicate that supervised machine learning approaches achieve superior performance compared to rule-based systems. For instance, only one fully rule-based system ranked better than the average of all 12 teams in the BioNLP'11 shared task (Kim *et al.*, 2011a). Opposing results have been reported for the BioCreative II.5 shared task for PPI extraction (Leitner *et al.*, 2010). In this competition, the best performing team implemented a rule-based system and achieved a F_1 of 42.9 %(Hakenberg *et al.*, 2010). In comparison, the machine learning system developed by Sætre *et al.* (2010) achieved a F_1 of 37.4 % on the same corpus. An interesting observation is that both systems reported F_1 results on an independent corpus, where the rule-based system performed approximately 7 percentage points worse than the machine learning based system. This indicates the need for robust relationship extraction methods, as well as a commonly accepted evaluation strategy to allow quantitative comparisons between different approaches.

2.5 Community-Wide Evaluation Efforts

In the last years, several attempts to a unified evaluation of biomedical relationship extraction have been carried out (Leitner *et al.*, 2010; Nédellec *et al.*, 2013; Segura-Bedmar *et al.*, 2013). Furthermore, there were individual efforts to consistently evaluate PPI extraction (Kabiljo *et al.*, 2009; Tikk *et al.*, 2010, 2013). This section introduces important gold standard corpora, describes obstacles for unbiased evaluations, and gives a broad overview of related work for relationship extraction. One important result of these individual attempts is that performance of rule-based systems is on a par with machine learning methods when simulating a more realistic use-case by cross-corpus evaluation.

Corpora

Corpora are manually annotated collections of texts where domain experts label relevant information, *i.e.,* those facts that should be extracted by an information extraction

system. The five most commonly used protein-protein interaction corpora are AIMed (Bunescu *et al.*, 2005), BioInfer (Pyysalo *et al.*, 2007), HPRD50 (Fundel *et al.*, 2007), IEPA (Ding *et al.*, 2002), and LLL (Nédellec, 2005). Pyysalo *et al.* (2008a) converted all five corpora into one unified format preserving the "greatest common factor" between corpora (*i.e.,* untyped and undirected binary relations). Some basic statistics about the converted corpora are available in Table 2.2.

These corpora have been annotated by different annotators using different annotation scopes. For instance, AIMed and HPRD50 focus on human genes, while LLL contains annotations for Bacillus subtilis. IEPA is the only corpus annotated with chemicals. The authors focused on a set of 16 different chemicals, while other annotators used the result of named entity recognition as preprocessing (*i.e.,* HPRD50). Only AIMed and BioInfer performed exhaustive manual annotation of genes and these two corpora also contain dispersed entities like "BRCA1/2", where BRCA2 is annotated as "BRCA" and "2". Further differences are the annotation of words stating the interaction (*e.g.,* bind, up-regulate, ...), negations, or the direction of an interaction.

A characteristic of AIMed is that proteins such as "TNFα-receptor" are annotated as two overlapping entities "TNFα-receptor" and "TNFα". Gold standard annotations for the sentence "AIMed.d32.s269" are shown in Figure 2.10. The figure shows that the string "erythropoietin (EPO) receptor" is annotated as three entities, namely the complete phrase but also the entailed words "erythropoietin" and the corresponding abbreviation "EPO". It is noteworthy, that the problem of granularity in named entity annotation leads to debatable annotations. For instance, in Figure 2.10 the annotators marked an interaction from "erythropoietin" and "EPO" to the abbreviation of the underlying entity "EPOR". Kim *et al.* (2010) identified nested entities as a burden in relation extraction, as the nested entities have identical context (and almost identical feature vectors), but sometimes different annotations.

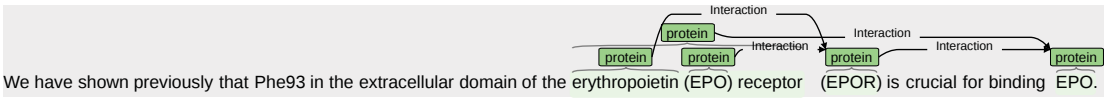
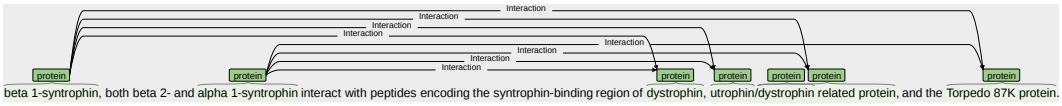
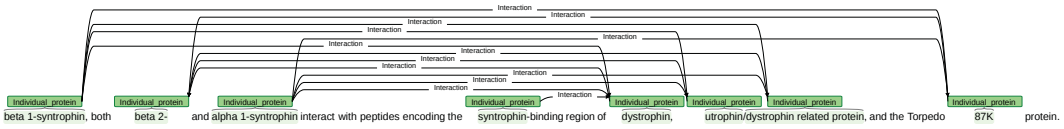


Figure 2.10: Gold standard annotation for the AIMed sentence AIMed.d32.s269. Please note that the string "erythropoietin (EPO) receptor" is annotated as three entities. Visualization performed using brat (Stenetorp *et al.*, 2012).

Differences between annotation guidelines can also be found by visualizing one of the few documents which are contained in more than one corpus. In total we found five documents shared between AIMed and BioInfer and one document shared between AIMed and HPRD50. Differences between AIMed and BioInfer are exemplified for one sentence in Figure 2.11. Annotations differ in the amount of annotated entities (BioInfer seems to be more complete) and in annotation boundaries for the last entity mention "Torpedo 87k protein".



(a) AIMed annotation



(b) BioInfer annotation

Figure 2.11: Annotation variants for the same sentence in AIMed and BioInfer (PubMed article PMID:8576247).

Corpus	Sentences	Positive pairs	Negative pairs
AIMed	1955	1000	4834
BioInfer	1100	2534	7132
HPRD50	145	163	270
IEPA	486	335	482
LLL	77	164	166

Table 2.2: Basic statistics of the 5 commonly used PPI corpora.

Comparability of reported results

A great number of publications dealing with the extraction of protein-protein interaction from text reported results on the previously introduced corpora. Although these methods use the same corpora for evaluation, results are often not directly comparable. A list of identified mistakes has been assembled by Pyysalo *et al.* (2008b) and the remainder of this section discusses some of the most important findings:

Parameter tuning Published results are usually derived by 10-fold cross-validation. One problem, already discussed in Section 2.3.1, is the exhaustive exploration of parameter space leading to overoptimistic performance estimations. In the context of PPI extraction it usually remains unknown how many experiments have been performed to achieve the presented results. Only few publications discuss the impact of different parameter settings on the overall performance of the presented system.

Entity blinding Another problem is the use of entity names as features for relationship extraction. This leads to a data leakage problem, as some corpora focus on a rather small range of named entities and the learner is prone to memorize previously seen interaction pairs. Such features are clearly not advisable as this affects generalization abilities on unseen entity pairs. To this end it is recommended to blind named entities using a generic string (*e.g.,* entity).

Level of cross-validation Sætre *et al.* (2007) showed that instance-wise cross-validations in contrast to document-wise cross-validation leads to overoptimistic performance estimates. The reason is that protein pairs mentioned within the same sentence, although having highly similar feature vectors, can end up in different cross-validation folds. The authors report a performance overestimation of approximately 10–20 %. This problem can be tackled by performing document-wise cross-validation, where instance from one document will always end in the same fold. Another important observation made by the authors is that the reported size of evaluation corpora differs between publications. This is due to different preprocessing steps (*e.g.,* the removal of overlapping named entities as used in different publications).

Impact of evaluation corpus Pyysalo *et al.* (2008a) estimated that the corpus choice affects F_1 on average by 19 percentage points and that the different positive/negative interaction pair distribution of the five benchmark corpora accounts for about half of the diversity in PPI extraction performance.

Evaluation criteria The performance of a classifier also depends on the evaluation criterion. Giuliano *et al.* (2006) introduced two different evaluation approaches: First, "One Answer per Occurrence in the Document" requires that every instance has to be classified correctly. Whereas the "One Answer per Relation in a given Document" requires only one correct answer for any occurrences of the same protein pair. The latter criterion leads to an increase of approximately 5 percentage points on the AIMed corpus, without any changes on the RE-method.

This section discussed several problems in the evaluation of relationship extraction methods, where we focused on findings made in the domain of protein-protein interactions. Several of these aspects show the importance of robustness. For instance, approaches without entity blinding perform well on corpora focusing on a small set of named entities, but performance will most likely decrease when applied to text focusing on arbitrary proteins. Similar problems occur when using extensive parameter tuning or improper evaluation scenarios (*e.g.,* instance-wise cross-validation). Most importantly, Pyysalo *et al.* (2008a) argued that a large proportion of corpus diversity is not due to semantic differences between PPI corpora, but rather because of different class distributions. On large unannotated text repositories, such as MEDLINE, the expected class distribution for most relationship types remains unknown.

2.5.1 Results for Protein-Protein Interaction Extraction

This subsection discusses the advances in protein-protein interaction extraction over the last 10 years. To allow comparability to a certain degree we only consider systems which were evaluated on the AIMed corpus. AIMed is considered as the de-facto standard for the evaluation of PPI methods and several publications use this corpus to evaluate their relation extraction approach. Experimental results shown in Table 2.3 are collected from the respective publication and Figure 2.12 shows the performance trend over the

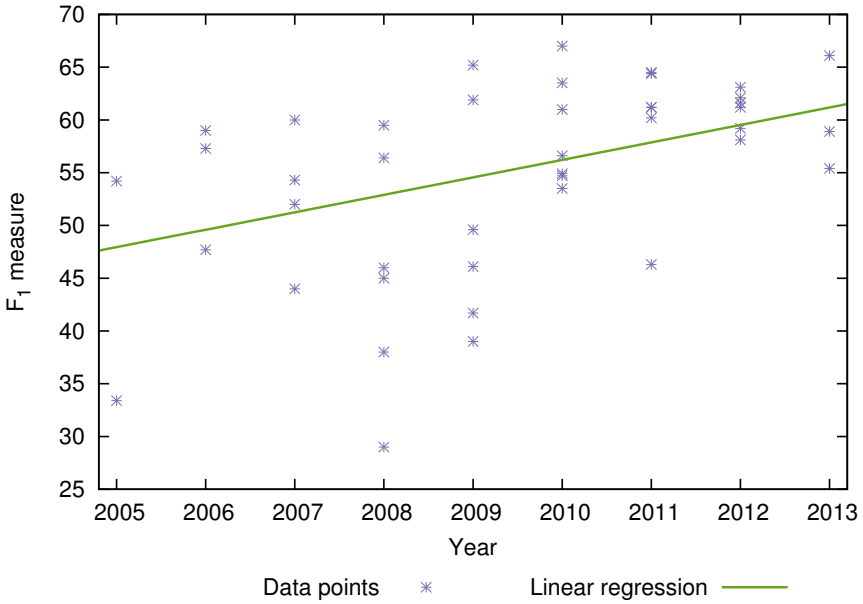


Figure 2.12: Performance for PPI extraction on AIMed over the last eight years. Linear regression has been fit on this data with an estimated yearly increase of 1.7 percentage points in F_1. Data extracted from Table 2.3.

last years. In various cases, properties like the number of performed cross-validation experiments, type of evaluation (instance or document-wise cross-validation), or entity blinding remain unknown from the original paper. For these reasons, results need to be considered with caution as differences in evaluation strategy have substantial impact on reported results (see previous section). The final part of this subsection describes selected approaches in chronological order. In this review we focused on the following aspects: novelty (*e.g.,* publications introducing new features), performance, and robustness (*e.g.,* cross-corpus performance).

System	Precision	Recall	F_1	Pattern-based	Machine learning	Constituency parser	Dependency parser	Abstract level CV
Yakushiji *et al.* 2005	33.7	33.1	33.4	✓	✓	✓	✗	✓
Bunescu and Mooney 2005b ❶	65.0	46.4	54.2	✗	✓	✗	✗	✓
Mitsumori *et al.* 2006	54.2	42.6	47.7	✗	✓	✗	✗	✓
Yakushiji *et al.* 2006	71.8	48.4	57.3	✓	✓	✓	✗	?
Giuliano *et al.* 2006	60.9	57.2	59.0	✗	✓	✗	✗	✓
Fundel *et al.* 2007 ❷	40.	50.	44.	✓	✗	✗	✓	—
Sætre *et al.* 2007	64.3	44.1	52.0	✗	✓	✗	✓	✓
Katrenko and Adriaans 2007	45.0	68.4	54.3	✗	✓	✗	✓	?
Erkan *et al.* 2007	59.6	60.7	60.0	✗	✓	✗	✓	✗
Baumgartner *et al.* 2008 ❸	61.0	9.1	29.0	✓	✗	✗	✗	—
Fayruzov *et al.* 2008b	—	—	38.	✗	✓	✗	✓	✓
Fayruzov *et al.* 2008a ❹	41.	50.	45.	✗	✓	✗	✓	?
Van Landeghem *et al.* 2008	49.	44.	46.	✗	✓	✗	✓	✓
Airola *et al.* 2008	52.9	61.8	56.4	✗	✓	✗	✓	✓
Miyao *et al.* 2008	54.9	65.5	59.5	✗	✓	✓	✓	✓
Fayruzov *et al.* 2009	—	—	39.0	✗	✓	✗	✓	✗
Nguyen *et al.* 2009 ❺	53.4	34.2	41.7	✗	✓	✓	✓	✓
Palaga 2009	49.4	44.7	46.1	✗	✓	✗	✓	✓
Strötgen *et al.* 2009	48.5	50.8	49.6	✓	✗	✗	✓	—
Miwa *et al.* 2009b	58.7	66.1	61.9	✗	✓	✗	✓	✓
Miwa *et al.* 2009a	60.0	71.9	65.2	✗	✓	✗	✓	✓
Niu *et al.* 2010	70.2	43.2	53.5	✓	✓	✓	✓	✓
Liu *et al.* 2010a	63.4	48.8	54.7	✗	✓	✓	✓	✓
Miwa *et al.* 2010	—	—	54.9	✓	✓	✓	✗	✓
Kim *et al.* 2010	61.4	53.3	56.6	✗	✓	✗	✓	✓
Katrenko *et al.* 2010	69.1	54.6	61.0	✗	✓	✓	✓	✗
Li *et al.* 2010	60.5	68.3	63.5	✗	✓	✗	✗	✓
Choi and Myaeng 2010	72.8	62.1	67.0	✗	✓	✓	✗	✓
Chowdhury *et al.* 2011	56.9	39.0	46.3	✗	✓	✗	✓	✓
Zhang *et al.* 2011b	54.9	68.5	60.2	✗	✓	✗	✓	✓
Zhang *et al.* 2011a	63.4	59.3	61.2	✓	✓	✗	✓	?
Bui *et al.* 2011	55.3	68.5	61.2	✓	✓	✓	✗	✓

Yang *et al.* 2011	57.7	71.1	64.4	✗	✓	✓	✓	✓
Li *et al.* 2011	—	—	64.5	✗	✓	✗	✗	✓
Qian and Zhou 2012	59.1	57.6	58.1	✗	✓	✓	✓	✓
Chowdhury and Lavelli 2012a	58.1	60.3	59.2	✗	✓	✗	✓	✓
Chowdhury and Lavelli 2012b	64.4	58.3	61.2	✓	✓	✓	✓	✓
Chowdhury and Lavelli 2012c	63.3	59.9	61.6	✓	✓	✓	✓	✓
Lee *et al.* 2012	54.9	71.3	62.1	✓	✓	✓	✓	?
Zhang *et al.* 2012	62.2	65.6	63.1	✗	✓	✓	✓	✓
Simões *et al.* 2013	49.4	64.1	55.4	✗	✓	✗	✓	✓
Tikk *et al.* 2013	58.0	61.1	58.9	✗	✓	✗	✓	✓
Raja *et al.* 2013	80.3	56.1	66.1	✓	✗	✓	✗	—

Table 2.3: Overview of published results for protein-protein interaction extraction on AIMed. Constituency parsing is only marked when the method works on the constituency parses and is not used as intermediate step (*e.g.,* when transformed to a dependency parse). A dash in abstract wise cross-validation indicates that no cross-validation has been performed, which is usually the case for pure pattern based approaches. Results are presented for up to one decimal place, when available. Publications not explicitly mentioning the level of cross-validation are indicated using a question mark. For five approaches, AIMed results are not mentioned in the original publication and have been extracted elsewhere: ❶ Results from Sætre *et al.* (2007); ❷ results from Pyysalo *et al.* (2008a); ❸ results from Kabiljo *et al.* (2009); ❹ results from Van Landeghem *et al.* (2008); ❺ results from Chowdhury *et al.* (2011).

To the best of our knowledge, Yakushiji *et al.* (2005) present the first PPI extraction approach evaluated on AIMed. They automatically construct patterns on the output of the predicate argument structure parser Enju by extracting the smallest set of predicates including the two interacting proteins. Predicate argument patterns are then matched against the held-out test split using document-wise cross-validation.

Bunescu and Mooney (2005b) present the subsequence kernel, building the foundation for several following relationship extraction approaches. The instance representation works on the surface level only and is an extension of the string kernel presented by Lodhi *et al.* (2002). The method works by splitting sentences into three fragments, where the idea is that one of these three fragments contains all information in order to express a relationship. The three fragments are defined as:

- Fore-Between: All words before and between the two entity names
- Between: All words between the two entity names
- Between-After: All words between and after the two entity names

For two fragments the kernel counts the number of shared n-grams, where fragment size is normalized using a constant factor λ. This kernel is than applied to calculate pairwise similarity for all three fragments between two instances. It is noteworthy that the authors published their results only as precision-recall graphs using the "One Answer per Relation in a given Document" evaluation criterion. Therefore, the estimated F_1 of 54.2 % is highly optimistic.

Giuliano *et al.* (2006) propose the shallow linguistic kernel, which is, despite its straightforwardness, still one of the best performing methods. The kernel is defined as the sum of the "global context" kernel and the "local context" kernel. The global context kernel is based on the subsequence kernel proposed by Bunescu and Mooney (2005b), where the number of common sequences is counted. The local context kernel uses morphologic (capitalization, punctuation, numerals, ...) and shallow linguistic features (*i.e.,* POS-tags and lemmas) from the token left and right of the protein pair.

Another important concept has been defined by Bunescu and Mooney (2005a) in the context of newspaper relationship types on the Automated Content Extraction (ACE) corpus. The authors introduce the shortest path hypothesis, stating that the relation establishing information is almost exclusively concentrated on the undirected shortest path between two named entities. Tokens located on the shortest path are transformed into a feature vector using information about token, part-of-speech, and entity type. This work was very influential for the following years and the shortest path assumption is frequently used when working with dependency parses.

Erkan *et al.* (2007) extract the shortest dependency path between two entities and implement cosine- and edit-distance as kernel functions. The advantage of these kernel functions, in comparison to Bunescu and Mooney (2005a), is that they can be used to calculate a similarity between dependency paths exhibiting different length. The authors observe that edit similarity achieves better performance than cosine similarity. Edit distance accounts for word order which potentially leads to better performance. The authors employ experiments using transductive learning, where the held-out test data (without class labels) is used during the training phase. This new optimization problem turns out to be NP-hard and therefore several approximation algorithms have been proposed (Zhu, 2008). The authors use the heuristic introduced by Joachims (1999), keeping positive to negative ratio between labeled and unlabeled data constant. Using transductive learning, the authors report a small increase in F_1. However, approaches aiming to replicate this result were unsuccessful and mostly reported a sharp increase in training time (Tikk *et al.*, 2010).

Fundel *et al.* (2007) present RelEx, the first PPI extraction system using patterns defined on the dependency parse. In a first step the dependency tree is compressed into a "noun phrase chunk" tree, where noun phrases are represented as a single node. The system uses three rules on the compressed tree to identify protein-protein interactions. The shortest path for each protein pair detected by the three rules is then scanned

for negation words. Protein pairs containing a negation word on the shortest path are subsequently removed. Cause and effector entity are identified using a simple heuristic. Originally, the authors evaluated RelEx on the HPRD50 corpus only, but Pyysalo *et al.* (2008a) reimplemented the system and evaluated it on additional corpora (including AIMed).

Van Landeghem *et al.* (2008) introduce the concept of vertex and edge walks on dependency parses. First, the authors use the shortest path assumption to reduce the dependency parse and then build 3-grams on the dependency path (originally referred to as walks), differentiating between vertex and edge walks. Vertex walks consist of two tokens and the connecting dependency, whereas edge walks contain the in- and out-going dependency and the common token. Walk features are extracted on part-of-speech and token level and this information is substituted with bag-of-words features for the whole sentence. Feature space is condensed using feature selection. Most interestingly, the authors evaluated the impact of instance-wise cross-validation in contrast to document-wise cross-validation. They observed a decrease of 16 percentage points in F_1 (62 % to 46 %) on AIMed after switching from instance to document cross-validation. This observation is very close to the decrease of 17.5 percentage points (69.5 % to 52.0 %) reported by Sætre *et al.* (2007) using a different extraction approach on the same corpus. These large differences support again the need for a common evaluation practice and robust methods in order to enable better comparability of approaches.

Baumgartner *et al.* (2008) utilize the concept of "Direct Memory Parsing" (Riesbeck, 1986) for the recognition of protein-protein interactions using the Open-DMAP framework (Hunter *et al.*, 2008). Open-DMAP is a general purpose ontology-driven information extraction framework for template matching. The authors developed 67 rules based on the shallow surface level of the sentence implemented in a Backus-Naur inspired grammar. An example of such a pattern[2] is:
`{interact-noun} {prep} (the)? [interactor1] and (the)? [interactor2]`,
where elements in brackets represent named entities and words in braces can be replaced with a number of alternative symbols. The rules are originally developed in the context of BioCreative-II, where the system ranked first with an F_1 of 29 %. The same rules have later been evaluated on the AIMed corpus where the system achieved a precision and recall of 61.0 % and 9.1 % respectively (Kabiljo *et al.*, 2009).

Airola *et al.* (2008) motivate the all-path-graph kernel, by showing that the shortest path sometimes misses important clue words for expressing a relationship. To define the kernel, the authors introduce two graphs. The first graph equals the linear order of tokens, where each neighbor is connected to its predecessor. Each token is associated with the information whether the token is located before, in-between, or after the protein pair. The second graph is the dependency parse, where edge labels (dependency types)

[2] Patterns are available at `http://sourceforge.net/projects/opendmap/files/supplementalPatterns/`

are represented as separate nodes. Edge weights on the dependency parse are set to 0.3, except for edges located on the shortest path where the weight 0.9 is used. Both graphs are then represented in terms of an adjacency matrix $\mathcal{A}$, where $\mathcal{A}_{i,j}$ contains the weight of the edge from node i to node j. The Neumann Series allows to calculate the sum of weights of all possible paths lengths. The similarity (kernel function) between two graph instances is then defined as the sum of shared paths for all nodes in a graph (Gärtner *et al.*, 2003).

Miwa *et al.* (2009b) aggregate several levels of information. First, they derive different syntactic sentence representations by using two different parsers. The authors then define three kernel functions which are applied on both parse representations. The first kernel evaluates words occurring before, in-between, and after the protein pair in question. The second kernel function counts the number of common subtrees contained in the shortest path. The last kernel utilizes the previously mentioned all-path-graph kernel. The final similarity function sums the normalized output for all three kernels and both parse tree representations.

Miwa *et al.* (2009a) use a domain adaptation technique to exploit annotations distributed in multiple corpora. As previously discussed, the five corpora have substantial differences in class distribution and annotation guidelines, which impedes the incorporation of potentially useful but different resources. The supervised machine learning component is based on the previously defined system using different parsers and kernel functions (Miwa *et al.*, 2009b). In contrast to their previous work the authors use explicit feature representations and a 2-norm soft margin SVM. These modifications increase F_1 by 2.7 percentage points on AIMed. For domain adaptation the approach introduces the two concepts target and source corpus. Target represents the currently investigated main corpus and source represents the union of the remaining four corpora. The goal is to learn a classifier for the target domain, utilizing annotations from target as well as source domain. Following the approach of Schweikert *et al.* (2008), the authors train a support vector machine using different cost parameters for target (C_t) and source corpus (C_s). This reformulates the original soft margin SVM problem (see Formula 2.4) into:

$$\arg\min_{\mathbf{w},b,\xi} \frac{1}{2}\|\mathbf{w}\|^2 + C_t \sum_{i=1}^{n} \xi_i + C_s \sum_{i=1}^{m} \xi_i \tag{2.16}$$

For PPI extraction this technique performs well on small corpora but provides only little improvement on AIMed and BioInfer. For AIMed the best increase of 1.2 percentage points in F_1 is observed when using only IEPA as additional source corpus. It is noteworthy that F_1 slightly decreases for other combinations, raising the question about knowing in advance which source corpus should be used. Furthermore, the presented domain adaptation approach assumes that the target domain is known. In real world scenarios it is often unclear whether for example AIMed reflects the target domain better than BioInfer. In Section 4 we present a different domain adaptation strategy avoiding the need of a previously defined target domain.

Miwa *et al.* (2010) perform sentence simplification to remove potentially misleading information. The authors manually defined seven rules based on the output of a head-driven phrase structure grammar parser to resolve linguistic properties such as apposition, copula, and coordination. Rules are applied recursively until the sentence remains unchanged. After each iteration a new parse tree is generated. Subsequently, the authors apply their previously defined relation extraction system (Miwa *et al.*, 2009b). The authors report that different rules have different effects on the five evaluation corpora. Furthermore, the authors curated 241 protein pairs and showed that the meaning was changed in only five cases. The application of all seven rules has the highest impact on HPRD50, IEPA, and LLL, whereas on AIMed and BioInfer only a subset of rules is useful. The authors state that the strategy helped to increase performance on all five datasets. However, a considerable decrease from 61.9 % to 54.9 % can be observed on AIMed, if results are directly compared with the originally published outcome (Miwa *et al.*, 2009b).

Niu *et al.* (2010) evaluate the impact of a variety of features. These features encompass information about the entity (*e.g.,* overlap with other entities or location in the text), words occurring in proximity to the entity (comparable to the subsequence kernel), manually defined patterns, constituency tree, and the dependency tree. A particularly interesting feature is the inclusion of a mixture model predicting if the sentence contains at least one protein-protein interaction. This feature alone increases F_1 by 2.3 percentage points.

Zhang *et al.* (2011b) make use of the "hash graph kernel", which has been originally proposed elsewhere (Hido and Kashima, 2009). This kernel iteratively propagates information between adjacent nodes trough the graph. To this end, each vertex is converted into a binary array representation using a mapping function. The size of the binary array has to be much larger than the alphabet of all nodes. In this work, the authors use arrays with a length of 24 bits. The mapping function is bijective but the numeric array representation is assigned randomly. For each vertex, the method calculates the neighborhood hash value by a series of bit shift and exclusive OR operations based on the directly connected neighbors. Repeating this step propagates information between neighboring nodes. Similar to Airola *et al.* (2008), the authors use two graph representations. One graph contains the dependency graph and the other graph contains the connected collocated tokens. Similarity between two graphs is derived as the sum of nodes with the same label (24-bit representation) divided by the sum of all nodes, where nodes located on the shortest path are given more weight than nodes located outside the shortest path. The authors briefly mention that too many iterations lead to a decrease in performance.

Bui *et al.* (2011) propose a hybrid approach where protein pairs are grouped according to their semantic properties. The authors define five syntactic forms to describe a protein-protein interaction in text. Protein pairs are grouped into these syntactic forms

using manually defined constituency-tree patterns. For each syntactic form the authors manually define an individual list of features and train a separate classifier. The previously defined patterns match 81.7 % of all interacting protein pairs on AIMed, defining the upper recall-boundary for the subsequent machine learning step. One advantage of this two-step strategy is that it filters many true negative instances, leading to a more balanced positive to negative ratio in all five datasets. Learning individual classifiers for each syntactic form leads to an increase of 6 percentage points in F_1.

Li *et al.* (2011) introduce a semi-supervised learning technique called "feature coupling generalization". The general idea is that some features are rarely (if ever) observed in the training data, but have a predictive value on the test data. To this end, the method searches for frequent co-occurrences of sparse features with so called class-distinguishing features (CDF) in unlabeled data. CDFs are selected by calculating χ^2-values between features and class labels on the annotated training corpus. Sparse features frequently co-occurring with a CDF are then generalized to higher-level features. This methodology leads to a reduced feature set. The approach is applied to three different classification tasks: named entity recognition, protein-protein interaction extraction, and text classification. Compared to the original features, feature coupling generalization increases F_1 by approximately 3.1 percentage points on AIMed. The proposed system performs extraordinarily well, given the fact that the system requires no syntactic parses.

So far we saw that the shortest path assumption is often used to restrict feature generation for dependency parses. Interestingly, there is no clear tendency for the representation of constituency parses. Zhang *et al.* (2008) explore different representations and conclude that the shortest enclosed parse performs best on the ACE corpus. The shortest enclosed parse contains all constituents starting from the lowest common subsumer between two named entities. This representation has been used by Choi and Myaeng (2010) and achieves superior results on AIMed. Qian and Zhou (2012) introduce a novel representation of constituency trees by incorporating information from the dependency parse. The authors extract the shortest constituency path and enrich it with all tokens entailed in the shortest dependency path. The authors compare their constituency representation with four other representations and show that it outperforms all others on the five corpora. Interestingly, the authors also use the shortest-enclosed constituency path, which leads to strikingly worse results (47.1 %) than the reported measures from Choi and Myaeng (2010) (67.0 %). Similarly, Yang *et al.* (2011) report a F_1 of 50.1 % when using the shortest enclosed path alone. We also implemented the shortest-enclosed path strategy and achieve a F_1 of 48.6 % which is between the results reported by Yang *et al.* (2011) and Qian and Zhou (2012).

Chowdhury and Lavelli (2012a) compare two different kernels using different parsers as inputs and different preprocessing strategies. An interesting observation made by the authors is that the type of entity blinding can have substantial impact on performance estimates. In this study they compare two blinding strategies, where either the first character is written uppercase or the whole token is written in uppercase. The former

strategy outperforms the all-uppercase strategy with up to 2 percentage points in F_1 on AIMed. One of the reasons for the worse performance values are wrongly assigned POS tags possibly leading to incorrect parse trees. A more practical solution to alleviate this problem is the application of entity blinding subsequently to syntactic parsing. However, this strategy has not been evaluated by the authors. Another interesting observation is that the "removal of parenthetical comments containing no entities" improves the result on some corpora.

Chowdhury and Lavelli (2012c) describe three simple rules to remove entity pairs which are likely not participating in a relationship. For instance, they remove pairs where both entities refer to the same mention. This strategy also incorporates information about abbreviation mentions. After removal of presumably non-interacting instances, the authors train the relation extraction method explained in Chowdhury and Lavelli (2012b). The rules reduce the amount of negative instances by approximately 20 %, but remove only 0.6 % of all positive instances. This leads to a more balanced positive to negative ratio while retaining most positive instances. The authors observe an increase in F_1 on four corpora but almost no effect for BioInfer.

This subsection presented the progress in PPI extraction over the last years for 43 systems (see Table 2.3). Due to differences in evaluation criteria, cross-validation type, and parameter optimization strategy (refer to Subsection 2.5) results have to be considered carefully. In early years two frequently used concepts have been presented: The first concept assumes that the relation is either mentioned by the phrase before, between, or after both entity mentions. The other concept introduces the shortest path hypothesis, stating that the relation establishing information is almost exclusively concentrated on the undirected shortest path between two entities. Most of the presented systems build features or define kernel functions based on one or even both concepts.

For PPI extraction systems, dependency graphs are clearly favored over constituency trees. For instance, 31 systems use dependency graphs but only 17 systems use constituency trees for relationship extraction. This trend is supported by results of Tikk *et al.* (2010), who concluded that dependency graphs are more suitable for relationship extraction than constituency trees. 10 systems incorporate both grammar types, in order to exploit the complementing syntactic information.

A clear tendency towards machine learning systems can be observed. Only four systems are purely rule-based and the remaining 39 rely, at least partially, on machine learning methods. An interesting observation is that 9 systems combine rule-based and machine learning based approaches. This ranges from simple but effective rules, like a protein and the directly mentioned abbreviation are unlikely to interact with each other (Chowdhury and Lavelli, 2012c) to more complex rules separating sentences by their semantic properties (Bui *et al.*, 2011).

Although there is an emerging need for robust relationship extraction only a small fraction of the presented approaches actually evaluates cross-corpus performance. Publications performing cross-corpus experiments usually observe a dramatic drop in F_1. To the best of our knowledge the only domain-adaptation study in the context of protein-

protein interactions has been performed by Miwa *et al.* (2009a). However, this work requires the a priori definition of a target domain, which is often impractical in real-world applications. In contrast we will discuss ideas to increase extrinsic performance without the explicit definition of a target domain.

2.5.2 Comprehensive Benchmarks

All previously presented systems have been evaluated on the AIMed corpus, but important details such as the number of performed experiments, parameter optimization strategy, entity blinding, or type of cross-validation often remain unknown. The high number of published work concerning relationship extraction increases the difficulty of selecting an appropriate algorithm. Due to the relatively high amount of time needed to run third party software (Ballardini *et al.*, 2011), extensive comparisons with other tools are rarely performed. This subsection describes recent efforts to a uniform evaluation of different relationship extraction approaches.

The effect of eight different parsers and five different parse tree representations (*e.g.,* PTB, CoNLL, ...) has been evaluated in the context of protein-protein interactions by Miyao *et al.* (2008). The authors utilize the kernel function from Sætre *et al.* (2007) to evaluate the impact of different parser/format combinations. Performance differences between diverse parsers are comparably small, but PTB (the standard format for constituency parses) performs worse than the other representations. This is underpinned by the observation that performance generally increases when converting PTB to any dependency tree format (*e.g.,* Stanford or CoNLL dependencies). Overall performance can be increased by combining the output of two kernels using different parsers or parse tree representations.

Several relationship extraction tasks, such as protein-protein interactions, are often evaluated in a closed setting where named entities are already annotated. This strategy allows a fair comparison of different relationship extraction algorithms, but the derived results cannot be extrapolated to real applications. This leads to overoptimistic performance estimations due to the missing named entity recognition step. The reason is that named entity recognition is imperfect and the relationship extraction algorithms are not trained to deal with incorrect or missing named entities. Kabiljo *et al.* (2009) evaluated the impact of named entity recognition for relationship extraction. For a realistic setting, the authors benchmark different named entity and relationship extraction tools. In 10 out of 15 cases, a decrease of at least 15 percentage points F_1 can be observed when removing named entity annotations. An important observation by the authors is that sophisticated tools generally perform worse than simple keyword-based approaches when switching to the more realistic evaluation scenario.

We have previously compared nine different kernels in the context of protein-protein interaction extraction (Tikk *et al.*, 2010). Four kernels use the syntax tree representation of sentences, but calculate different subtree representations (see Figure 2.6 for the different tree representations). Four kernels are based on the dependency graph and one kernel uses only shallow linguistic surface representation. All methods are evaluated on the five previously mentioned corpora using a unified evaluation strategy. This encom-

passes: the use of the same parses (if used by the individual method), document-wise 10-fold cross-validation, unified entity blinding, and coarse-grained parameter optimization. Similar to the results of Pyysalo *et al.* (2008a), we observe that the performance strongly depends on the evaluation corpus (see Figure 2.13 for results). Nevertheless, the methods introduced by Airola *et al.* (2008) and Giuliano *et al.* (2006) perform consistently well on all five corpora.

This work was later extended to compare 13 machine learning based methods (Tikk *et al.*, 2013). The main contribution is the analysis of instances wrongly classified by most methods. This leads to the identification of 521 negative and 190 positive instances with critical difficulty level. Subsequent analysis on critical, neutral, and easy instances reveals a correlation between difficulty level and sentence length, where positive instances are harder to identify in longer sentences and negative pairs are harder in shorter sentences. To counteract problems of individual methods, the authors build a majority voting combination for the three best performing methods raising F_1 by 2.7 percentage points on AIMed.

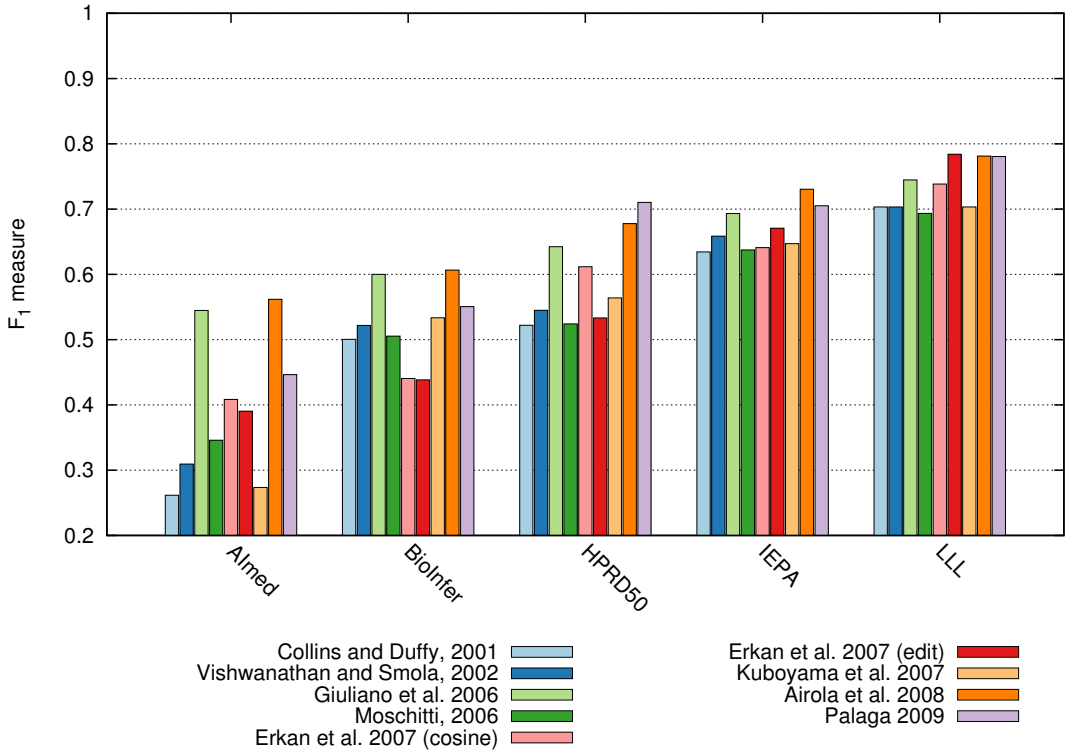


Figure 2.13: Comparison of nine relation extraction methods for five corpora. Results from Tikk *et al.* (2010).

2.5.3 Community Evaluation Efforts

Another well accepted approach for benchmarking systems are shared tasks. In advance to a specific conference, problems are proposed and training data is provided by the organizers. Participants are invited to develop a system to solve the given task. Test data (without gold standard annotations) is usually published several month later and participants must then submit solutions within a given time frame. This time slot is usually a few days long, to avoid manual intervention by individual participants. Shared tasks have a long tradition in information extraction, as for instance for the "message understanding conferences" held between 1987 and 1997. The advantage of shared tasks is that they are evaluated on previously unseen test data and therefore provide a more robust evaluation than conventional cross-validation. After the end of shared tasks, the annotations are ideally not publicly released. Researchers are allowed to provide a certain amount of submissions per day (*e.g.,* one run per day) to evaluate later developed systems. This strategy increases comparability of later approaches, as it avoids common pitfalls in evaluation. However, Kaufman *et al.* (2011) showed that shared tasks may still lead to overoptimistic estimates when subject to data leakage or cheating by participating teams. For instance, the authors mention the INFORMS 2010 data mining challenge where more than 30 teams were able to map "blinded" test instances to the corresponding stock-data leading to almost perfect predictions of "future" stock price movements.

The probably most important community efforts in the biological domain are the BioCreative conferences and the BioNLP shared tasks. Over the last years, BioCreative covered several different topics ranging from gene name recognition (Hirschman *et al.*, 2005; Krallinger *et al.*, 2008), over identification of PPI relevant articles and the extraction of protein-protein interactions (Leitner *et al.*, 2010), to interactive demonstrations of text mining systems (Arighi *et al.*, 2011).

The BioNLP shared task was introduced by Kim *et al.* (2009). A characteristic feature of the BioNLP shared task, compared to binary relationship extraction, is the definition of fine-grained event types. Events generally consist of a trigger word (expressing the relation) and an arbitrary number of arguments. A particularity is the definition of nested events, which take another event as an argument.

Due to its success the challenges were repeated twice with different data (Tsujii *et al.*, 2011; Nédellec *et al.*, 2013). In 2011 the organizers prepared five main event extraction tasks, differing in text types, event types, and domain. The ambitious goal was to provide several different corpora to evaluate domain adaptation capabilities of different systems. However, only one team successfully participated in all maintasks and subtasks (Björne *et al.*, 2012). The theme of 2013 was to support the construction of knowledge bases. To this end, the organizers defined six different tasks relevant for knowledge base construction. The organizers also provided a larger body of supporting resources, encompassing syntactic parses and results from multi-purpose named entity recognition tools.

Another shared task worth mentioning are the two drug-drug interaction extraction challenges (Segura-Bedmar *et al.*, 2011a, 2013). This task shares several similarities with the protein-protein interaction extraction task, as drug-drug interactions are defined as

undirected binary relations between two entities in the same sentence. We participated in both shared tasks using an ensemble of different classifiers. The drug-drug interaction task and our contribution will be described in more detail in Chapter 3.

3 Ensemble Methods for Relationship Extraction

This chapter discusses the usability of ensemble learning techniques for relationship extraction. Ensembles aggregate the output of several heterogeneous classifiers in order to reduce the risk of accidentally choosing an inappropriate single classifier. For this reason ensemble methods are generally considered to increase robustness. We choose the domain of drug-drug interactions (DDIs) in order to compare ensemble methods over individual classifier performance. The work presented in this chapter has been originally developed in the context of the SemEval 2013 shared task[1] and ranked second among eight participants on an unseen test corpus (Segura-Bedmar *et al.*, 2013).

3.1 Ensemble Learning

Ensemble learning refers to the process of combining several individual classifiers in order to build a stronger classifier. The methodology is inspired by the process of human decision making, where individuals ask the opinion of several people in order to come to a final decision. Previous community competitions showed that ensemble learning helps to achieve better performances than relying on one single method (Kim *et al.*, 2009; Leitner *et al.*, 2010).

An important property of ensembles is that they increase robustness by decreasing the risk of selecting a bad (or miscalibrated) classifier (Polikar, 2006). For instance, it is straight forward to obtain several different classifiers on a data set. This can be accomplished by learning different classification algorithms (*e.g.,* SVM, Naïve Bayes, or logistic regression) or by different parameter settings (*e.g.,* different soft-margin misclassification costs) on the same dataset. Assume that some of these classifiers exhibit, according to 10-fold cross-validation, similar F_1. However, performance on the unseen test set may considerably vary among the different classifiers. In such cases, it can be advantageous to combine learned classifiers in order to reduce the risk of randomly choosing a particularly bad classifier. The aggregated result does not necessarily outperform all individual classifiers on test-data but is more robust in terms of performance.
Ensemble learning theory typically distinguishes two combination types:

1. In *classifier selection*, the goal is to train several classifiers (or experts) for different areas. During prediction, the algorithm selects the most suitable classifier for every provided test instance according to some formal criteria (*e.g.,* depending on feature

[1] Joint work with M. Neves, T. Rocktäschel, I. Solt, and U. Leser

allocation). This approach can be compared with a general practitioner sending patients to the respective specialist according to the disease pattern.

2. In *classifier fusion*, one is interested in combining the prediction of several "weak" classifiers to form a "strong" forecast. This approach is also known as the " wisdom of the crowd".

In this thesis, we will examine two well established classifier fusion techniques (majority voting and stacking) in order to achieve higher robustness on unlabeled test data. We focused on classifier fusion techniques, as they received much more attention than classifier selection (Kuncheva, 2004, Chapter 3.2.1).

3.1.1 Majority Voting

A frequently used ensemble technique is majority voting, where the class with most votes (provided by individual classifiers) is predicted. The advantage of majority voting can be explained by the following example: Assume we have a binary classification problem with T classifiers. Each classifier exhibits an uncorrelated individual error rate of p. Independent classification error rates lead to different prediction errors and are a requirement for ensemble learning, as fully correlated classifiers provide no additional information. The probability of observing exactly k misclassifications can be determined using the binomial distribution shown in Formula 3.1:

$$f(T; n, p) = \binom{T}{k} p^k (1-p)^{T-k} \tag{3.1}$$

Therefore, the misclassification probabilities using T uncorrelated classifiers can be estimated as:

$$P(H(\mathbf{x}) \neq y) = \sum_{k=T/2+1}^{T} \binom{T}{k} p^k (1-p)^{T-k} \tag{3.2}$$

The advantage of majority voting can be shown by the following example: Given a set of classifiers with individual error rates of $p = 1/3$ (equaling to an accuracy of $2/3$), the relation between increasing numbers of classifiers on expected accuracy is visualized in Figure 3.1. For instance, combining 21 individual classifiers by majority voting reduces the misclassifications rate by one order of magnitude from $1/3$ to almost $1/40$ (0.026).

3.1.2 Classifier Diversity

Majority voting is guaranteed to improve performance over individual classifiers given independent error rates and individual accuracies above 50 %. Unfortunately, classifiers often produce somewhat correlated output, generally leading to less impressive improvements. For this reason, a subgoal of ensemble learning is to build as many diverse, but still well performing, individual classifiers as possible.

Diversity between classifiers is often artificially introduced by building different data or feature subsets. For instance, this is implemented in bagging (Breiman, 1996), where

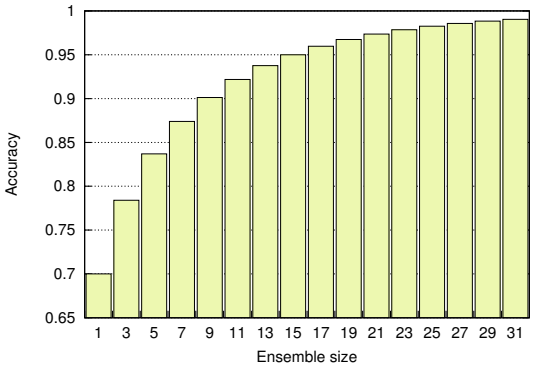


Figure 3.1: Expected accuracy using majority voting for different numbers of available classifiers. Individual predictors exhibit an uncorrelated error rate of 1/3.

different classifiers of the same type are learned on different bootstrap samples from the entire training data. Predictions from all classifiers are combined by majority voting to form a final decision. A related approach to introduce noise into the data is implemented in Random Forest (Breiman, 2001). Here, several decision tree classifiers are learned on feature spaces sampled from the original space. In this chapter, we will work with ensemble learning techniques, not relying on artificially introduced noise to build different classifiers.

3.2 Drug-Drug Interactions

Modern drugs are ubiquitous in most people's everyday life. As of 2008 almost 50 % of all American citizens took at least one prescribed drug (Gu *et al.*, 2010). Beside pharmaceutical medications, a wide variety of other drugs (such as nicotine, steroids, caffeine, or anti-aging products) are frequently used.

A drug-drug interaction (DDI) occurs when the effectiveness of one drug is influenced by the presence of another drug, typically due to simultaneous administration. Most known DDIs lead to an increase or decrease of drug effect in comparison to the isolated administration of one drug alone. For instance, sildenafil in combination with nitrates can cause a potentially live-threatening decrease in blood pressure (Cheitlin *et al.*, 1999). It is, therefore, crucial to consider potential DDI effects when co-administering drugs to patients. As the level of medication is generally raising all over the world, the potential risk of unwanted side effects, such as DDIs, is constantly increasing (Haider *et al.*, 2007). Specialized resources such as DrugBank already cover more than 4,800 drugs together with possible interactions (Knox *et al.*, 2011). These resources also cover interactions with fruits, which are generally considered to be healthy, but can have huge effect on metabolisation of drugs (Bailey *et al.*, 2013). Zwart-van Rijkom *et al.* (2009) observed at least one DDI for 27.8 % of all hospitalized patients in a dutch hospital. Additional studies indicate that available knowledge about dangerous drug combinations is not

sufficiently incorporated into the decision process and medical doctors require more computational assistance (Cavuto *et al.*, 1996; Smalley *et al.*, 2000).

3.2.1 DDI-2013 Task Description

At the moment, knowledge about DDIs is not always optimally integrated into the decision making process. Meanwhile, up-to-date DDIs repositories are required to develop computational assistance tools. In 2013, the SemEval Task 9 organizers presented the DDI-challenge targeting a uniform and fair evaluation of participating teams for two tasks related to DDIs identification in texts. The presented corpus incorporated the DDI-challenge 2011 corpus (Segura-Bedmar *et al.*, 2011b). Previous annotations have been revised as named entities were automatically recognized using text-mining components (Herrero-Zazo *et al.*, 2013). The first task (Task 9.1) focused on named entity recognition of drug mentions and the second task (Task 9.2) consisted of the extraction of drug-drug interactions. Entities in Task 9.1 were categorized into four different classes, which are *brand*, *group*, *drug*, and *drugN*. Details about the four subtypes can be found in the task's annotation guideline. An example for Task 9.1 annotations is shown in Figure 3.2(a).

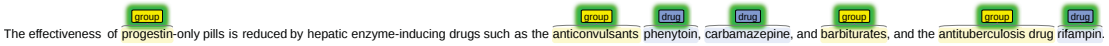
The second assignment (Task 9.2) encompassed the extraction of undirected binary relations between co-occurring drugs mentioned in the same sentence. This definition is very similar to the greatest common factors defined in the context of protein-protein interaction corpora (see Section 2.5). In contrast to the previous DDI-challenge 2011, four different DDI-subtypes (*advise*, *effect*, *mechanism*, and *int*) have been proposed:

- *Advise* is assigned to DDIs where a recommendation or advise for co-administration is mentioned. For example, when mentioning the amount of time (*e.g.,* one hour) by which co-administration should be delayed.
- *Effect* is used to label DDIs describing the pharmacological effect, such as an increase in drug toxicity.
- *Mechanism* describes how an interaction takes effect.
- *Int* is used to label interactions that do not provide sufficient information for further sub-classification.

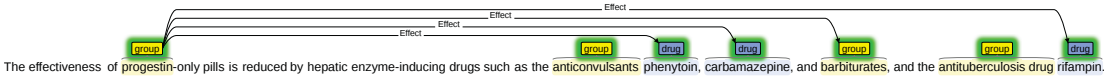
An example for Task 9.2 annotations is shown in Figure 3.2(b).

3.3 Methods

Binary relationship extraction is frequently tackled as a pair-wise classification problem, where all $\binom{n}{2}$ possible entity-pair combinations in a sentence are classified as interacting or not (see Subsection 2.4.2). To account for the four different DDI-subtypes, the problem definition here could be translated into a multi-class classification problem between all co-occurring entities.



(a) Annotations for Task 9.1



(b) Annotations for Task 9.2

Figure 3.2: Example annotations from the SemEval Task 9 training corpus.

Contrary to that, we follow a two step coarse-to-fine grained classification strategy. First, we detect general drug-drug interactions regardless of subtype (*i.e.,* advise, effect, mechanism, and int) using a multitude of heterogeneous relationship extraction methods. Robustness on unseen text is increased by aggregating the output of individual classifiers using ensemble learning techniques. Second, interactions detected in the first step are re-classified into one of the four possible DDI categories. This re-classification step will be referred to as relabeling. The complete workflow is depicted in Figure 3.3.

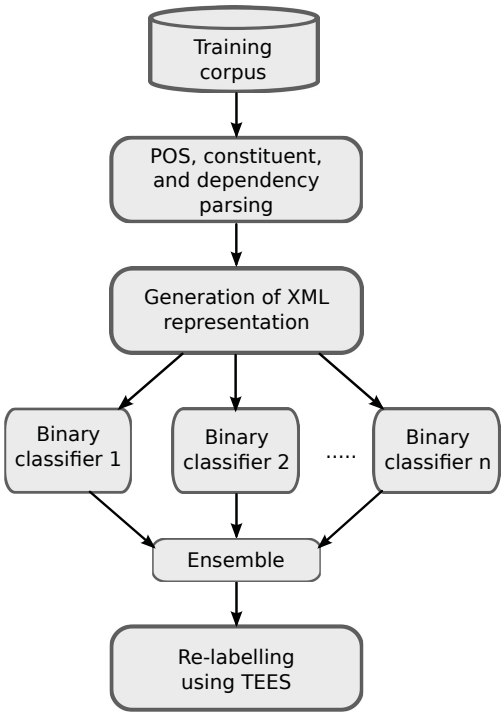


Figure 3.3: Workflow developed for SemEval 2013 task 9.2.

3.3.1 Preprocessing

The organizers provided annotations for two different text resources. The majority of annotations was provided for 572 DrugBank articles, where each article concentrated on a specific drug. The remainder was provided for 142 MEDLINE abstracts. More details about the two training corpora can be found in Table 3.1. The distribution of DDI subclasses are shown in Figure 3.4 for both corpora. The figure indicates a different subclass distribution between the two corpora. The most pronounced difference can be found for subtype "advise" with observed frequencies of 21.6 % and 3.4 % for DrugBank and MEDLINE respectively.

Each article is provided as one XML file, containing annotations for sentence boundaries, named entities, and drug-drug interactions. We syntactically enrich this information by applying the following preprocessing steps: Sentences are parsed using a constituency parser (Charniak and Johnson, 2005) with a self-trained re-ranking model augmented for biomedical texts (McClosky, 2010). Resulting constituent parse trees are converted into dependency graphs using the Stanford converter (De Marneffe *et al.*, 2006). We transform the original XML file into an augmented XML encompassing part-of-speech tags, constituency-, and dependency-parse tree information.

Corpus	Documents	Sentences	Pairs		
			Positive	Negative	Total
DrugBank	572	5,675	3,788	22,217	26,005
MEDLINE	142	1,301	232	1,555	1,787

Table 3.1: Basic statistics of the DDI training corpus shown for DrugBank and MEDLINE separately.

3.3.2 Relation Extraction Methods

Entities are blinded by replacing the entity name with a generic string (*e.g.,* "sildenafil" becomes "drug"). Entity blinding is necessary in order to increase robustness on unseen entity pairs (see Subsection 2.5). It is important to note that disabling entity blinding often increases performance in intrinsic evaluations (such as cross-validation) because the classifier tends to memorize co-occurring drug-names. However, entity blinding is highly advised for realistic evaluation scenarios. For instance, when searching for novel interactions that are not contained in the training set. For the DDI-2013 evaluation corpus we observe that 233 of all 5716 (4.1 %) drug pairs also occurred in the training corpus. In order to improve generalizability we refrained from disabling entity blinding. We previously estimated the impact of entity blinding in the context of the drug-drug interaction challenge 2011 and observed an increase of 1.7 percentage points in F_1 without entity blinding (Thomas *et al.*, 2011d). It is noteworthy that some participants explicitly used drug names as features or used other features such as "this drug-drug

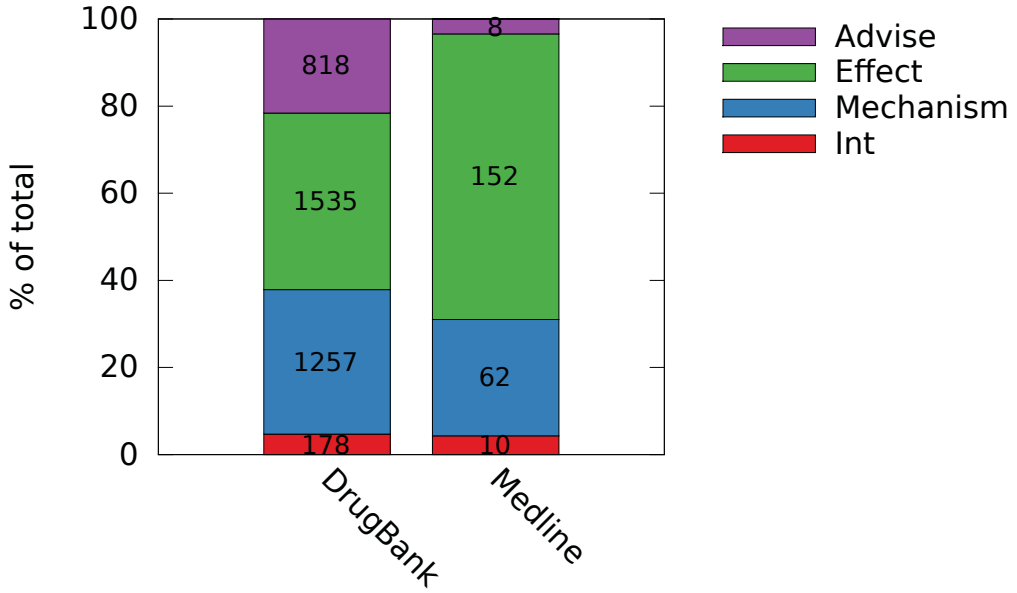


Figure 3.4: Distribution of DDI subclasses in percent for both training corpora. Numbers inside the boxes represent the actual number of observed instances for that specific subclass.

interaction is already contained in DrugBank". As explained, such features are likely to increase intrinsic performance, but can mislead classifiers in real world applications where the goal is the detection of unknown drug-drug associations.

We will base our study of ensemble's on a variety of selected relationship extraction methods provided by the relationship extraction framework of Tikk *et al.* (2010). These are the all path graph kernel (APG) (Airola *et al.*, 2008), shallow linguistic kernel (SL) (Giuliano *et al.*, 2006), subtree kernel (ST) (Vishwanathan and Smola, 2002), subset tree kernel (SST) (Collins and Duffy, 2001), and spectrum tree kernel (SpT) (Kuboyama *et al.*, 2007). We excluded the k-band shortest path spectrum kernel (kBSPS) (Palaga, 2009) as the classifier showed unrobust performance in previous competitions (Solt *et al.*, 2010; Thomas *et al.*, 2011d). Due to high runtime requirements we excluded the partial tree kernel (PT) (Moschitti, 2006) after a couple of preliminary experiments. Description of the individual methods can be found in Section 2.5.1.

In addition to the PPI framework, we employed the general purpose relationship extraction tool "Turku Event Extraction System" (TEES) (Björne *et al.*, 2011), a self-developed feature based classifier which is referred to as SLW, and a customized version of the case-based reasoning system Moara (Neves *et al.*, 2009):

- TEES considers features related to the tokens (*e.g.,* part-of-speech tags), dependency chains, dependency path N-grams, entities (*e.g.,* entity types) and external resources, such as hypernyms in WordNet. As the authors of TEES also participated in the DDI challenge, TEES will be explained in more details in the related

work section of this chapter.

- SLW is inspired by the shallow linguistic kernel (Bunescu and Mooney, 2005b; Giuliano *et al.*, 2006). For each protein pair we extract n-grams over sequences of arbitrary features (*e.g.,* POS-tags, morphological and syntactical features) to describe the global context of an entity pair. Furthermore, we generate features describing the local context of entities (*i.e.,* words left and right from the entities in question). The data set divides drugs into four different subtypes (*i.e.,* Brand, Group, Drug, DrugN). We observed in the training data that some co-occurring entity subtypes are more likely to interact (*e.g.,* Brand and Group) than others (*e.g.,* Brand and Brand). This observation is captured by including the class names of both entity mentions into our feature representation. Prior probabilities for all co-occurring entity classes are shown in Table 3.2.

- Moara is a case-based reasoning system for the extraction of relationships and events. Interaction pairs are converted into cases composed of the following features: The subtype of both entities (*e.g.,* Brand and Group), the part-of-speech (POS) tag of tokens between the two drugs, the POS tags of the shortest dependency path between the two drugs, and the lemma of the non-entity tokens of the shortest dependency path derived from BioLemmatizer (Liu *et al.*, 2012). Lemmas matching the pharmacogenomic relationship (PHARE) ontology (Coulet *et al.*, 2011) are replaced by the respective category term. Case similarity is calculated by exact feature matching, except for the part-of-speech tags whose comparison is based on global alignment using insertion, deletion, and substitution costs as proposed by Spasic *et al.* (2005).

Entity1	Entity2	Interaction	Total	Percentage
Brand	Group	698	1,908	36.5 %
Brand	Drug	1,348	5,272	25.5 %
Drug	Group	2,110	13,066	16.1 %
Drug	DrugN	146	1,020	14.3 %
Group	DrugN	18	156	11.5 %
Drug	Drug	2,964	26,034	11.4 %
Group	Group	654	5,768	11.3 %
Brand	DrugN	2	30	6.6 %
DrugN	DrugN	28	466	6.0 %
Brand	Brand	28	1,604	1.7 %

Table 3.2: Relationship between prior probabilities for drug-drug interactions depending on the two entity subtypes (*entity1* and *entity2*). Column *interaction* specifies the number of observed drug-drug interactions and *total* represents the number of co-occurring entities with this specific subtype.

3.3.3 Ensemble Learning

In this work we combine the output of several classifiers by using two different well established ensemble algorithms *i.e.,* majority voting and stacked generalization.

Majority voting

The first strategy combines binary predictions (interaction or not interaction) of individual classifiers (*i.e.,* APG, SL, TEES, ...) by majority voting (see Subsection 3.1.1). In this setting all predictors receive one, equally important, vote and the most frequently predicted class is returned. Voting ties are avoided by choosing only uneven (*i.e.,* $\{3, 5\}$) classifier combinations.

Stacked generalization

Stacked generalization (or stacking) is an alternative ensemble algorithm. Stacking learns a meta-classifier (also called level-1 classifier) on the output of the individual classifiers (Wolpert, 1992). The individual relationship extraction algorithms (*i.e.,* APG, SL, TEES, ...) are referred to as level-0 classifiers $\mathcal{L}_1 \dots \mathcal{L}_N$. Training the meta-classifier follows a slightly modified k-fold cross-validation strategy. Similar to regular CV each level-0 classifier is trained on a subset $\mathcal{D} \setminus \mathcal{D}_i$ and applied on the remaining dataset $\mathcal{D}_i$. The individual predictions of all level-0 classifiers on $\mathcal{D}_i$ form, together with the correct class label, new training instances. The meta classifier is then learned on the new training instances assembled over all k folds. In difference to majority voting, the meta-classifier uses the distance to the hyperplane (except for Moara) from the level-0 classifiers as feature and not only the binarized predictions. This allows stacking to incorporate the confidence of each classifier to build a final decision.

3.3.4 Relabeling

The previously described ensembles are used to predict DDIs regardless of the four different interaction subtypes (advise, effect, mechanism, and int). This binary untyped relationship extraction complies with the partial match evaluation criterion defined by the competition organizers and is the usual evaluation scheme in the context of protein-protein interactions. To account for DDI subtypes, previously identified DDIs are relabeled into one of the four possible subtypes. To this end, we use TEES multi-class classification capabilities to assign the most probable DDI subtype to previously identified interactions. This means that the ensemble predicts the general presence of an interaction between two drugs and TEES subsequently determines the subtype.

3.4 Results

This section covers results for untyped relationship extraction of individual methods and combinations using different ensembles. Results are shown for 10-fold cross-validation as well as for the blinded evaluation corpus. We further study the impact of merging

the two individual text sources (DrugBank and MEDLINE) when training individual classifier. Finally, we evaluate re-labeling performance using TEES.

3.4.1 Cross-Validation

In order to compare the different approaches, we performed document-wise 10-fold cross-validation on the training corpora. All approaches use identical cross-validation splits to ensure comparability of the different classifiers. For APG, ST, SST, and SpT we followed the parameter optimization strategy defined by Tikk *et al.* (2010). For TEES and Moara, we selected parameters using a coarse parameter selection strategy. For SL and SLW, we used the default SVM parameters.

The organizers provide two source corpora annotated using the same annotation guidelines. The MEDLINE corpus consists of scientific abstracts, whereas the DrugBank corpus consists of specific paragraphs extracted from the DrugBank website. By reading some of the annotations of both corpora we observed for DrugBank repeating phrases such as "Co-administration of DrugX with DrugY leads to ...", indicating that DrugBank is more homogeneous than MEDLINE. Similarly, Chowdhury and Lavelli (2013b) reported a much stronger use of the cue words "increase" and "decrease" in DrugBank, indicating that DrugBank has a more typed vocabulary.

We therefore compared some of the corpus specific aspects such as average sentence length or number of entities per sentence. We also calculated the normalized Shannon entropy (Shannon, 1948) which is defined for observing an arbitrary probability distribution of N tokens. Shannon entropy quantifies the asymmetry in the observed probability distribution, where $H(X) = 1$ represents uniform probability distribution (*i.e.,* all token are identically distributed) and $H(X) = 0$ shows a fully localized probability distribution; (*i.e.,* observing only the identical token).

Results of this analysis are shown in Table 3.3. Significance between all sentence wise characteristics is derived using a two sided Mann-Whitney U-test (Mann and Whitney, 1947). The null hypothesis is that the median between the two characteristics is zero. Differences for all characteristics, except length of shortest dependency path, are significant (significance level $\alpha = 0.05$). This change is mostly pronounced for the amount of entities per sentence, resulting in a higher number of co-occurring drug pairs in DrugBank compared to MEDLINE (7.4 pairs vs. 3.5 pairs).

This analysis leads to the question if the two corpora are similar enough to be considered as the same domain or not. We investigated this question by following two different cross-validation strategies: First, performance of relationship extraction is estimated for each corpus individually (DrugBank and MEDLINE). This is implemented by following a regular document-wise 10-fold cross-validation for each corpus. In the second experiment, cross-validation data is complemented by data from the other corpus. For instance, we perform regular cross-validation on DrugBank, but add the whole MEDLINE corpus to the training instances. This strategy allows us to estimate the impact of additional, but potentially different text sources for both corpora. Performance of individual methods and different majority voting ensembles for DrugBank and MEDLINE are shown in Table 3.4 and 3.5 respectively.

Characteristic	DrugBank	MEDLINE	p-value
Avg. no. of tokens per sentence	25.5	26.9	$5.0 \cdot 10^{-3}$
Avg. no. of entities per sentence	3.5	2.8	$5.9 \cdot 10^{-16}$
Avg. no. of tokens between two entities	9.5	8.8	$4.6 \cdot 10^{-6}$
Avg. no. of tokens on shortest path	3.8	4.0	0.2671
Normalized entropy $H(X)$	0.56	0.66	—

Table 3.3: Statistics of different characteristics for both DDI training corpora. Only sentences with at least one entity pair are considered. p-values are derived using Mann-Whitney U-test.

	Method	Regular CV				Combined CV			
		P	R	F_1	AUC	P	R	F_1	AUC
Individual Classifier	SL	61.5	79.0	69.1	92.8	62.1	78.4	**69.2**	93.0
	APG	77.2	62.6	**69.0**	91.5	75.9	59.8	66.7	91.6
	TEES	77.2	62.0	**68.6**	87.3	75.5	60.9	67.3	86.9
	SLW	73.7	60.0	65.9	91.3	73.4	61.2	**66.6**	91.3
	Moara	72.1	55.2	**62.5**	—	72.0	54.7	62.1	—
	SpT	51.4	73.4	60.3	87.3	52.7	71.4	**60.6**	87.7
	SST	51.9	61.2	**56.0**	85.4	55.1	57.1	**56.0**	86.1
	ST	47.3	64.2	54.2	82.3	48.3	64.3	**54.9**	82.7
Majority Voting	SL+SLW+TEES	76.1	69.9	**72.7**	—	75.9	65.3	70.1	—
	APG+SL+TEES	79.3	69.9	**74.2**	—	79.2	65.4	71.5	—
	Moara+SL+TEES	79.9	69.6	**74.2**	—	79.6	65.1	71.6	—
	Moara+SL+APG	81.4	70.6	**75.5**	—	81.3	70.3	75.3	—
	APG+Moara+SL+SLW+TEES	84.0	68.1	**75.1**	—	83.7	64.2	72.6	—
	APG+SpT+TEES	76.8	68.0	**72.1**	—	77.1	63.4	69.6	—
	APG+SpT+SL	68.7	74.8	71.5	—	69.7	73.8	**71.6**	—

Table 3.4: Cross-validation results for DrugBank. Regular CV is training and evaluation on DrugBank only. Combined CV referes to suplementing DrugBank with instances from MEDLINE. Higher F_1 between these two settings are indicated in boldface for each method. Single methods are ranked by F_1.

	Method	Regular CV				Combined CV			
		P	R	F_1	AUC	P	R	F_1	AUC
Individual Classifier	TEES	70.7	36.0	44.5	82.2	59.6	46.5	**51.4**	84.9
	SpT	37.8	38.6	34.6	78.6	42.3	55.3	**47.1**	80.4
	APG	46.5	44.3	42.4	82.3	38.1	62.2	**46.4**	82.8
	SST	31.3	37.7	31.8	74.1	36.7	61.7	**44.9**	79.5
	SL	43.7	40.1	38.7	78.9	34.7	67.1	**44.7**	81.1
	SLW	58.0	14.3	20.4	73.4	50.1	38.0	**42.0**	82.4
	Moara	49.8	31.9	37.6	—	45.6	43.2	**41.9**	—
	ST	25.2	43.8	30.1	70.5	36.1	48.3	**39.8**	74.2
Majority Voting	SL+SLW+TEES	73.6	29.0	37.6	—	55.2	52.7	**53.1**	—
	APG+SL+TEES	60.7	37.9	43.4	—	49.9	62.4	**54.3**	—
	Moara+SL+TEES	68.0	33.0	42.2	—	62.1	55.5	**57.4**	—
	Moara+SL+APG	57.7	36.7	42.4	—	48.3	60.9	**52.8**	—
	APG+Moara+SL+SLW+TEES	73.3	28.3	36.8	—	60.6	54.4	**56.5**	—
	APG+SpT+TEES	58.5	37.4	41.7	—	57.5	59.2	**57.1**	—
	APG+SpT+SL	48.3	39.9	40.0	—	43.6	64.3	**51.0**	—

Table 3.5: Cross-validation results for MEDLINE. Regular CV is training and evaluation on MEDLINE only. Combined CV referes to suplementing MEDLINE with instances from DrugBank. Higher F_1 between these two settings are indicated in boldface for each method. Single methods are ranked by F_1.

CV results for the DrugBank corpus (Table 3.4) show no clear effect when using MEDLINE as additional training data. By adding MEDLINE instances during the training phase we observe an average decrease of 0.3 percentage points (pp) in F_1 and an average increase of 0.7 pp in AUC. The small increase in AUC indicates that additional data helps to learn a slightly better discrimination between the two classes, but most classifiers are unable to select the optimal threshold value. This is reflected by the minor decrease in F_1. The strongest impact of additional MEDLINE training data on DrugBank can be observed for APG with a decrease of 2.3 pp in F_1. For almost all ensembles (with the exception of APG+SpT+SL) we observe superior results when using only DrugBank as training data. Interestingly, this effect can mostly be attributed to an average increase of 3.3 pp in recall, whereas precision remains fairly stable between ensembles using DrugBank solely and those with additional training data from MEDLINE.

In contrast, for MEDLINE all algorithms clearly benefit from additional training data with an average increase of 9.8 pp and 3.6 pp for F_1 and AUC respectively. For the ensemble based approaches, we observe an average increase of 13.8 pp for F_1 using the additional annotations from DrugBank. These results indicate that MEDLINE gains from additional out-domain data, whereas the effect on DrugBank is unclear. One possible explanation is the difference in corpus size, where MEDLINE constitutes almost 15 times less training instances than DrugBank. It is possible that corpora with

sufficient training instances are more likely to be distracted by out-domain information than small corpora with few annotations.

Cross-validation results for both corpora show significantly better F_1-estimates for DrugBank in comparison to MEDLINE (Wilcoxon signed-rank test; p = 0.003906). Also differences in efficiency of relationship extraction algorithms can be observed. When ranking the different methods by F_1 and calculating rank-correlation between the two different corpora, we observe a very weak correlation (Kendall's $\tau = 0.286$, p= 0.4). In other words, machine learning methods show varying performance-ranks between the two corpora. This difference is most pronounced for SL and SpT, with four ranks difference between DrugBank and MEDLINE. Additionally, documents come from different sources and it is tempting to speculate that there might be a certain amount of domain specificity between DrugBank and MEDLINE sentences. Without further experiments it remains unclear if differences in overall performance and performance rank are due to domain specific effects or due to different amounts of training instances.

3.4.2 Relabeling

Performance of relabeling is evaluated by performing 10-fold CV on the training set using the same splits as in previous experiments. Note that this experiment is solely performed on positive instances in order to estimate separability of the four different DDI-subtypes. Results are shown in Table 3.6.

Type	Pairs	Precision	Recall	F_1
total	3,119	78.6	78.6	78.6
effect	1,633	79.8	79.1	79.4
mechanism	1,319	79.8	79.2	79.4
advise	826	77.3	76.4	76.9
int	188	68.5	80.9	74.1

Table 3.6: Performance estimation for relabeling DDIs. Pairs denotes the number of instances of this type in the training corpus.

The DDI-relabeling capability of TEES is very balanced with F_1 measures ranging from 74.1 % to 79.4 % for all four DDI subclasses. This is unexpected since classes like "effect" occur almost ten times more often than other classes like "int" and classifiers often have problems with predicting minority classes.

3.4.3 Performance on the Test Set

For the SemEval 2013 competition participants were allowed to provide three individual submissions. At that time we had no results using stacking (for results see Subsection 3.4.4) and therefore submitted results for three majority voting ensembles. For Run 1 we used Moara+SL+TEES, for Run 2 we used APG+Moara+SL+SLW+TEES and for Run 3 we used SL+SLW+TEES. These ensembles have been selected as they

generally achieved good results according to 10-fold cross-validation. All classifiers, except APG, have been retrained on the combination of MEDLINE and DrugBank using the parameters yielding the highest F_1 in cross-validation. For APG, we trained two different models: One model is trained on DrugBank only and another model is trained on the union of both corpora. The first model is applied on the DrugBank test set and the latter on the MEDLINE test set. For each of the three runs, 10-fold cross-validation results on the training corpus as well as official results on the test corpus are shown in Table 3.7.

Evaluation	Training									Test								
	Run 1			Run 2			Run 3			Run 1			Run 2			Run 3		
	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
Partial	78.7	67.3	72.6	82.9	66.4	73.7	75.2	67.6	71.2	84.1	65.4	73.6	86.1	65.7	74.5	80.1	72.2	75.9
Strict	65.7	56.1	60.5	70.0	56.0	62.2	63.0	56.7	59.7	68.5	53.2	59.9	69.5	53.0	60.1	64.2	57.9	60.9
-mechanism	61.8	49.7	55.1	68.1	50.0	57.7	59.2	50.3	54.4	72.2	51.7	60.2	74.9	52.3	61.6	65.3	58.6	61.8
-effect	68.8	57.9	62.9	71.8	57.6	63.9	66.1	57.4	61.5	63.7	57.5	60.4	63.6	55.8	59.5	60.7	61.4	61.0
-advise	64.6	60.5	62.5	68.2	59.7	63.6	61.1	61.5	61.3	73.3	53.4	61.8	74.5	55.7	63.7	69.0	58.4	63.2
-int	68.6	50.0	57.8	75.4	52.1	61.6	70.9	56.9	63.1	67.8	41.7	51.6	67.3	38.5	49.0	67.8	41.7	51.6

Table 3.7: Relation extraction results on the training and test set. Run 1 builds a majority voting on Moara+SL+TEES, Run 2 on APG+Moara+SL+SLW+TEES, and Run 3 on SL+SLW+TEES. Partial characterizes only DDI detection without classification of subtypes, whereas strict requires correct identification of subtypes as well.

On the blinded test set, our best submission (Run 3) achieves a F_1 of 75.9 % using the partial evaluation schema. This is slightly better than the performance estimates for DrugBank (shown in Table 3.4) and substantially better than estimates for MEDLINE (see Table 3.5). With F_1 measures ranging between 74 % to 76 % only minor performance differences can be observed between the three different ensembles. Run 3 outperforms the other two ensembles on MEDLINE and DrugBank in terms of F_1 (Segura-Bedmar *et al.*, 2013, Table 6–8), indicating a higher robustness.

When switching from partial to strict evaluation scheme an average decrease of 15 pp in F_1 can be observed. As estimated on the training data, relabeling performance is indeed very similar for the four different DDI-subtypes. Only for the class with the least instances (*int*), a larger decrease in comparison to the other three classes can be observed for the test set. In general, results for test set are on par or higher than results estimated by cross-validation. Approaches and results for the seven competing teams will be discussed in Subsection 3.6.

3.4.4 Stacked Generalization

After the end of the DDI challenge we performed additional experiments for the prediction of untyped DDIs using stacked generalization. Predictions from TEES have been ignored, as we observe a high variance in confidence values which negatively affects stacking. Two different meta-classifier are evaluated: Naïve Bayes and Logistic regression.

These two methods have been selected as they require no elaborate parameter tuning strategy and thus minimize the likelihood of over-fitting.

Many classifiers use an optimization objective different from the non-linear F_1 measure, leading to suboptimal performances for F_1. This can be compensated by learning the optimal classification threshold (w.r.t F_1) on the training set (Nan *et al.*, 2012). To this end, a trained classifier is reapplied on the previously used training set. For each classified training instance we obtain a tuple consisting of the actual class (*e.g.,* -1/+1) and a real valued number representing the classifier confidence (*e.g.,* class probabilities for Naïve Bayes). We then search the threshold achieving highest performance with respect to F_1 on the training set. This fairly simple strategy increases F_1 on average about 1.9 percentage points on the unseen DDI test data. Results of stacking are shown in Table 3.8. Several models outperform our results of 75.9 % (Run 3) for untyped DDI extraction. The best performing system outperforms majority voting by 1.1 percentage points. During cross-validation we observed that majority voting is easily distracted by less informative classifiers (*i.e.,* ST, SST, and SpT). For this reason we ignored these classifiers for majority voting during the DDI competition. In contrast, stacked generalization seems to be not affected by adding less informative classifiers, due to increased generalization capabilities. We therefore conclude that stacked generalization provides higher robustness in comparison to majority voting.

Classifier	Feature set	Precision	Recall	F_1
Naïve Bayes	SL/SLW/Moara	77.9	66.9	72.0
Naïve Bayes	SL/SLW/Moara/APG	**80.2**	74.2	**77.1**
Naïve Bayes	SL/SLW/Moara/SpT	77.8	69.5	73.4
Naïve Bayes	SL/SLW/Moara/APG/SpT	79.8	73.5	76.5
Naïve Bayes	SL/SLW/Moara/APG/SpT/ST/SST	72.2	**80.2**	76.0
Logistic Regression	SL/SLW/Moara	75.7	69.2	72.3
Logistic Regression	SL/SLW/Moara/APG	**80.3**	73.4	76.7
Logistic Regression	SL/SLW/Moara/SpT	76.3	70.9	73.5
Logistic Regression	SL/SLW/Moara/APG/SpT	79.0	75.1	**77.0**
Logistic Regression	SL/SLW/Moara/APG/SpT/ST/SST	78.2	**75.2**	76.7

Table 3.8: Performance on the blinded test set using stacking.

3.5 Conclusion

This chapter described an approach originally implemented in the context of the SemEval 2013 – Task 9.2 DDI extraction challenge. Our strategy builds on a cascaded (coarse-to-fine grained) classification strategy, where a majority voting ensemble of different methods is initially used to find untyped DDIs. Predicted interactions are subsequently relabeled into four different subtypes using a multi-class classifier.

DDI extraction seems to be a more difficult task for MEDLINE abstracts than for DrugBank articles. This behavior can be observed for all participating teams, where F_1 is approximately 26 percentage points lower for MEDLINE than DrugBank. In a preliminary experiment we observed only minor differences between the two corpora. For instance, we observe a higher amount of entities per sentence in DrugBank and a slightly lower word entropy. However, other aspects like distribution of negations, frequency of other named entities, or passives between the two corpora could be investigated as well (Cohen *et al.*, 2010). Another difference between the two corpora is the availability of more training data for DrugBank, which potentially effects classification performance.

We tested for domain specificity by performing cross-corpus experiments, *i.e.,* we trained a classifier on DrugBank, applied it on MEDLINE and vice versa. When training on MEDLINE and testing on DrugBank, we observe an average decrease of about 15 pp in F_1 in comparison to DrugBank in-domain CV results. For the other setting, we observe a lower decrease of approximately 5 pp in comparison to MEDLINE in-domain CV results. Especially the second result points toward some domain specificity between the two corpora. Similar results have been observed when merging training instances from both corpora, where F_1 usually decreases on DrugBank although we have a higher number of training instances. We therefore assume that transfer learning techniques, similar to the approach by Miwa *et al.* (2009a) (see Subsection 2.5.1), could further improve results on both corpora.

3.6 Related Work

This section gives a brief overview of all seven competing teams. All teams applied machine learning techniques, some of them complemented by rules in order to improve the result. Interestingly, purely rule-based approaches were not presented. Support vector machines were the dominating machine learning algorithms as all teams made use of this technique. Few teams additionally incorporated other machine learning algorithms. This points toward the prevalent opinion that pure rule-based approaches are cumbersome to develop and often fail to achieve state-of-the-art result.

An overview of techniques used by participating teams is shown in Table 3.9. The table indicates that six of eight approaches followed a coarse-to-fine grained classification strategy identifying general DDIs first, followed by a re-classification step to identify the specific DDI subclass. Several participants describe ideas to deal with the large imbalance between positive and negative instances. Most teams used the union of both corpora for training. No other team investigated the impact of complementing training instances stemming from the complementing corpus. Three teams (FBK-irst, WBI, and SCAI) incorporated ensemble learning techniques. It is noteworthy that these three teams achieved the top three ranks according to the partial evaluation scheme.

Performance of individual submissions for all eight participating teams is shown in Table 3.10. Our approach consistently achieved the second best performance for all evaluation settings (*i.e.,* strict, partial, and all four subclasses). The UTurku team used a modified version of TEES adapted to the specific problem. This modified version

ranked third and performed only 1.5 percentage points worse using strict evaluation than our approach. Interestingly, we outperform the same approach by a larger margin of 6.0 percentage points using partial evaluation. This indicates that our ensemble based strategy performs considerably well in predicting untyped DDIs, but fails to predict the subtypes for many correctly identified interactions.

System	Pattern	Machine learning	Constituency parser	Dependency parser	Coarse-to-fine	Ensemble
FBK-irst	✓	✓	✓	✓	✓	✓
WBI	✗	✓	✓	✓	✓	✓
TEES	✗	✓	✗	✓	✗	✗
NIL_UCM	✗	✓	✓	✗	✓	✗
UC3M	✗	✓	✗	✗	✓	✗
UWM-TRIADS	✓	✓	✗	✗	✓	✗
SCAI	✓	✓	✗	✓	✓	✓
UColorado_SOM	✓	✓	✗	✓	✗	✗

Table 3.9: Overview of techniques used by participating teams in the context of the SemEval 2013 (Task 9.2) challenge. Constituency parsing is only marked when the method works on the constituency parses and is not used as preprocessing step (*e.g.,* when transforming to a dependency parse).

FBK-irst

The best performing system, presented by Chowdhury and Lavelli (2013b), incorporates several ideas from their previous publications (Chowdhury and Lavelli, 2012c, 2013a). First, the authors use a classifier developed to identify informative sentences. Informative sentences are defined as sentences containing at least one DDI. Sentences classified as uninformative are immediately discarded from further predictions. Second, the authors define several heuristics to remove probable negative instances. For instance, if both entity mentions refer to the same entity (also incorporating abbreviations) the pair will be excluded from further processing as self-mediated drug interactions are extremely unlikely. Finally, the authors train three support vector machines on different feature/kernel spaces. The individual predictions are combined to one result by summation of the three predicted confidence values (*i.e.,* distance to the hyperplane). The impact of the different steps was not separately evaluated. For relabeling, the authors train four individual classifiers using a one-vs-all strategy. The class label with highest

Team	Run	Rank	Strict	Partial	Mechanism	Effect	Advise	Int
FBK-irst	1	3	63.8	**80.0**	**67.9**	**66.2**	**69.2**	36.3
	2	**1**	**65.1**	**80.0**	**67.9**	62.8	**69.2**	**54.7**
	3	2	64.8	**80.0**	62.7	**66.2**	**69.2**	**54.7**
WBI	1	6	59.9	73.6	60.2	60.4	61.8	51.6
	2	5	60.1	74.5	61.6	59.5	63.7	49.0
	3	**4**	60.9	75.9	61.8	61.0	63.2	51.0
UTurku	1	9	58.1	68.4	57.8	58.5	60.6	50.3
	2	**7**	59.4	69.6	58.2	60.0	63.0	50.7
	3	8	58.2	69.9	56.9	59.3	60.8	51.1
NIL_UCM	1	12	51.7	58.8	51.5	48.9	61.3	42.7
	2	**10**	54.8	65.6	53.1	55.6	61.0	39.3
UC3M	1	**11**	52.9	67.6	48.0	54.7	57.5	50.0
	2	21	29.4	53.7	26.8	28.6	32.5	40.2
UWM-TRIADS	1	17	44.9	58.1	41.3	44.6	50.2	39.7
	2	**13**	47.0	59.9	44.6	44.9	53.2	42.1
	3	18	43.2	56.4	44.2	38.3	53.7	29.2
SCAI	1	**14**	46.0	69.0	44.6	45.9	56.2	2.0
	2	16	45.2	68.3	44.1	44.0	55.9	2.1
	3	15	45.8	70.4	45.0	46.2	54.0	2.0
UCOLORADO_SOM	1	22	21.4	49.2	10.9	25.0	21.9	9.7
	2	20	33.4	50.4	36.1	31.1	38.1	33.3
	3	**19**	33.6	49.1	33.5	31.3	42.0	32.9

Table 3.10: Performance (F_1) of all eight teams participating in the SemEval 2013 task 9.2. Teams are ranked by overall performance using the *strict* evaluation scheme.

score is assigned to previously detected DDIs.

UTurku

Björne *et al.* (2013) use their previously developed tool TEES (Björne *et al.*, 2011) for the extraction of DDIs. The authors perform some changes to the original tool, such as ignoring *conj_and* dependencies when calculating the shortest path and by adding facts about named entities. This encompasses entity specific information about presence and category in DrugBank, as well as a boolean feature indicating if the currently investigated drug pair is already known to be interacting according to DrugBank. Additionally, the authors apply the multi purpose concept recognition tool MetaMap (Aronson and Lang, 2010) and incorporate identified concepts into the feature representation. According to the authors, the integration of DrugBank facts leads to an increase of 2 percentage points in F_1.

NIL_UCM

Bokharaeian and Diaz (2013) collect a large number of features on different levels. First, they assemble features such as words, stems, lemmas, and part-of-speech tags occurring before, between, and after the two drug entities. Features are extended by a bag-of-word representation of the whole sentence as well as the entity names. Additional features are extracted from the constituency parse and by detecting phrases with a negated scope using NegEx[2]. In order to reduce the original feature space the authors perform feature selection using information gain to derive the most informative features. Individual classifiers are learned for both training corpora. The authors compare the effect of multi-class classification versus a two step (coarse-to-fine) classification. Using the partial evaluation setting the two step strategy outperforms the multi-class classification strategy by a large margin of 6.8 percentage points in F_1.

UC3M

Sanchez-Cisneros (2013) propose a two step re-classification strategy using the shallow linguistic kernel. It seems that the author used the original implementation of the SL kernel, which returns only binary prediction values and not the distance to the hyperplane. Therefore conflicts can appear when more than one model predicts the respective subclass. These cases are resolved by using frequency information about the individual subclasses (*e.g.,* subclass *effect* is more frequent than *int*). Two individual runs have been submitted. The first run uses the shallow linguistic kernel as is, whereas the second run replaces named entities with the respective "Anatomical Therapeutic Chemical" (ATC) symbol. For the latter experimental setting, the authors observe a dramatic decrease of 13.9 percentage points in F_1.

[2] `https://code.google.com/p/negex/`

UWM-TRIADS

Rastegar-Mojarad *et al.* (2013) present a two-stage coarse-to-fine grained classifier. The authors discuss the problem of class imbalance. To this end the authors experimented with different resampling methods such as SMOTE (Chawla *et al.*, 2002), which deemed to be less successful as the usage of class specific soft-margin costs. The authors define a feature vector containing sentence-level features (which are identical for all instances in the sentences) and instance level features (which are different between different drug pairs in the same sentence). Similar to team "FBK-irst" the authors define a set of rules to remove probable non-interacting drug-pairs. These rules remove drug mentions having identical names (considering plural form, but not abbreviations). Furthermore, the authors remove drug pairs referring to the same pharmacologic drug class (*e.g.,* monoamine oxidase inhibitor) using information provided by the U.S. Food and Drug Administration[3]. Finally, drugs co-occurring as part of an enumeration are ignored.

SCAI

Bobic *et al.* (2013) build features for different linguistic levels. First, the authors generate lexical features for text before, between, and after the current entity pair. These features are mostly n-gram features with $n \in \{1, 2, 3\}$. The second class of features (syntactic features) are encoded as n-grams along the shortest dependency path between two entities. Semantic features are defined as negation words in close proximity, the entity class (*i.e.,* drug, drug_n, brand, group), prior observations of this entity pair (using the actual string value) in the training set, and if the pair refers to the identical named entity (incorporating abbreviations). The authors train three individual classifiers (SVM, Naïve Bayes, and voted perceptron) and evaluated the impact of two different ensemble strategies: *Majority* predicts an interaction when two or more classifiers support that claim. *Union* predicts an interaction if at least one classifier predicts an interaction. Relabeling is performed as post-processing step utilizing a manually compiled set of trigger words. Trigger words are mutually exclusive between the four interaction subtypes and are defined on sentence level. Therefore, the presented approach predicts always the same DDI-subtype for all DDIs recognized in a sentence. It is noteworthy, that although the system ranked 7^{th} in the competition, it achieves good results (3^{rd} rank) in the prediction of DDIs independently of subtype (partial evaluation), indicating that the rule-based re-labeling step requires some improvements.

UColorado_SOM

Hailu *et al.* (2013) build a one-vs-all classifier for each subclass using several features. Features encompass information on shallow level, such as token distance between the two co-occurring drugs, presence of other drugs between the drug pair, presence of manually defined interaction words, and bigrams of tokens. The authors also use features extracted from the dependency tree such as presence of interaction words and drug names on

[3] `http://www.fda.gov/ForIndustry/DataStandards/StructuredProductLabeling/ucm162549.htm`

the shortest path and features incorporated from the TEES system. One particularly interesting feature is the incorporation of OpenDMAP patterns developed in the context of BioNLP'09 challenge (Cohen *et al.*, 2009). An entry in the feature vector indicates if the two co-occurring drugs have been extracted by any of the provided OpenDMAP patterns.

Segura-Bedmar *et al.* (2014)

Segura-Bedmar *et al.* (2014) analyzed submissions from all eight teams participating in Task 9.2. Predictions of two individual runs have been analyzed using McNemar's test (McNemar, 1947). The analysis revealed that individual runs submitted by FBK-irst, WBI, UTurku, and SCAI did not show statistical significant differences between other runs from the same team. However, runs submitted by different teams showed significant difference with the exception of two statistical tests (UTurku vs. SCAI and UC3M vs NIL_UCM). The authors evaluated several majority voting ensembles. Interestingly, only one ensemble provided marginal improvement over the best participating team. All remaining ensembles lead to a decreased performance.

Bui *et al.* (2014)

Bui *et al.* (2014) showed that the five rules previously developed in the context of protein-protein interaction (see Bui *et al.* (2011) and Subsection 2.5.1) can be used to reliably identify negative training instance. This leads to a more balanced positive to negative class ratio in the training set. Depending on the corpus, the rules remove between 33.8 to 38.2 % of all negative instances but remove only a small fraction of positive instances (2.8 to 8.6 %). Remaining drug pairs are converted into a feature based representation involving shallow features.

4 Domain Adaptation using Self-Training

Over the last eight years we observed a constant increase in F_1 on AIMed (Figure 2.12). However, this improvement does not necessarily translate to better performance on data sets with potentially different text properties. Corpora are usually sampled from larger text collections (such as MEDLINE) by using some formal selection criteria (*e.g.,* containing specific key-words). Therefore, they often reflect only a specific subdomain of all available texts. This affects robustness of learned text mining components on arbitrary texts.

Robustness of a text mining component is assessed by so called *extrinsic* studies, where a model is evaluated on a corpus different from the training corpus. Extrinsic performance for protein-protein interaction extraction received fairly little attention, although several studies observed a strong performance decrease when switching from *intrinsic* cross-validation to *extrinsic* cross-learning (Pyysalo *et al.*, 2008a; Airola *et al.*, 2008; Tikk *et al.*, 2010). In this chapter, we analyze the impact of self-training, a semi-supervised learning strategy, to improve performance of protein-protein interaction extraction on texts with unknown characteristics[1].

4.1 Introduction

In Subsection 2.5 we discussed differences among the five most commonly used PPI corpora. For instance, they differ in annotation scope (*e.g.,* directionality, complex, negative interactions, ...), definition of a PPI (*e.g.,* permanent physical bindings versus transient contacts), and scientific subareas the corpus was built from (*e.g.,* human diseases). Pyysalo *et al.* (2008a) showed that the corpus choice affects F_1 on average by 19 percentage points and that the different positive to negative interaction pair distribution of the five benchmark corpora accounts for about half of the diversity of the performances of the PPI extraction approaches. Diversity of corpora also affects average sentence length, ranging from 26.5 to 35.8 words for HPRD50 and BioInfer respectively (Miwa *et al.*, 2010). To avoid corpus bias as much as possible, it is widely acknowledged that methods should be evaluated on all available corpora.

Machine learning methods for relationship extraction use existing corpora to train a model. To evaluate such a model, it is applied to new text. In text-mining, this setting is often simulated by 10-fold cross-validation within one corpus, which will be further referred to as *intrinsic* setting. 10-fold cross-validation actually suffers from the weakness that training and application data might exhibit very different characteristics in real world application, which is not properly reflected in this setting. This leads to

[1] Joint work with I. Solt, and U. Leser

unrealistic performance estimations and inadequately chosen models (Pan and Yang, 2010).

Methods to cope with these problems are gathered under the umbrella term "transfer learning". Research in transfer learning addresses, for instance, cases in which the training and test sets share the same feature space but not the same annotation schema (inductive transfer learning; ITL) or vice versa (domain adaptation; DA). In PPI extraction, features are usually shared in training and test sets, but class labels are not semantically equivalent across corpora.

Several approaches have been proposed to assess the effect of a shift in target domain. One may, for instance, train a classifier on one corpus (cross-corpus; CC) or on all available corpora except the evaluation corpus (cross-learning; CL). Figure 4.1 illustrates the different evaluation strategies (CC and CL are two extrinsic settings).

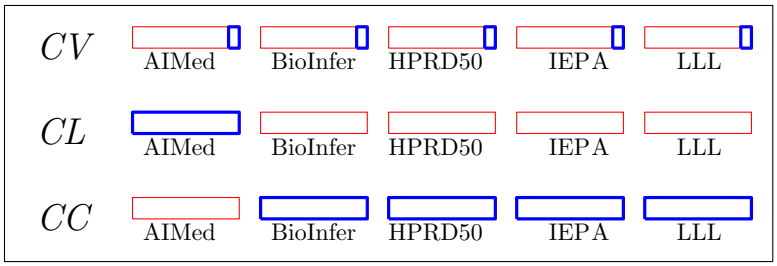


Figure 4.1: Evaluation settings typically used for PPI extraction. Thin red boxes represent training data and thick blue boxes the evaluation data. CV performs corpus-wise 10-fold cross-validation on document level. CL uses the union of all but one corpora for training and evaluates on the remaining corpus. CC uses a single corpus for training and separately evaluates on all remaining corpora.

To thoroughly assess the state-of-the art in PPI extraction, we previously compared nine relationship extraction methods in all three evaluation settings using a uniform tuning procedure (Tikk *et al.*, 2010). This benchmark clearly confirmed the large performance gap between intrinsic (within the same corpus) and extrinsic (across corpora) evaluations.

For instance, when training a PPI kernel on AIMed and evaluating it on BioInfer (CC), the performance in F_1 drops by roughly 12 pp compared to the CV result on BioInfer. For a given test corpus, *maximum* CC performance (optimal training corpus selection) superseded CL performance by up to 1-6 pp in F_1 (Tikk *et al.*, 2010, Tables 3 and 4). However, across all corpora, *average* performance in the CC setting was found to be inferior to that in the CL setting (see Figure 4.2). The optimal training corpus varies by test corpus, rendering CC evaluation less practical to predict performance on unlabeled documents. Altogether, although a certain performance drop was expected, the extent of this drop is surprisingly high (see Table 4.1) and confirms differences in annotation principles and text selection used for the different corpora. Note that similar problems have been observed in other domains, such as named entity recognition (Alex

et al., 2006; Wang, 2010) or constituent tree parsing (McClosky *et al.*, 2006b). As a consequence, it is hardly possible to give a sensible estimate on the performance one may expect from any of those methods on unseen text. Furthermore, it remains unclear how methods could benefit from training corpora of similar but differing annotation schemata, a common requirement for large-scale PPI extraction.

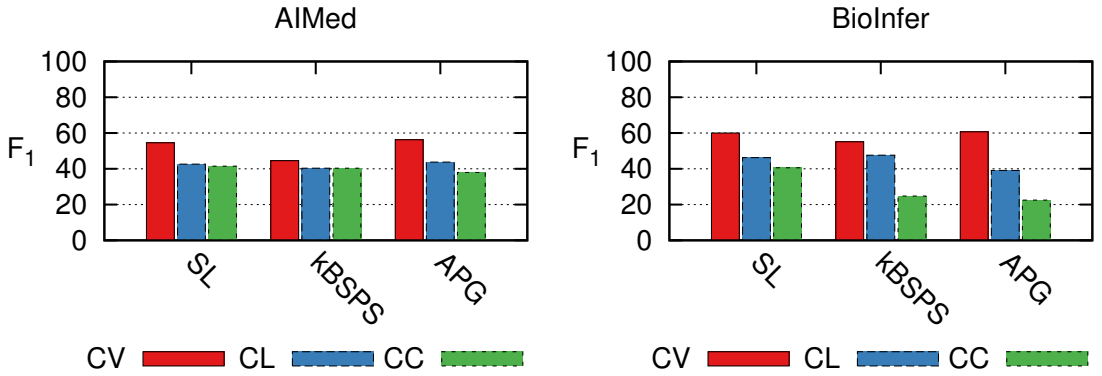


Figure 4.2: Comparison of three different relationship extraction methods using CV, CL and CC evaluation on AIMed and BioInfer. CC represents average performance over all four possible training corpora. Data from Tikk *et al.* (2010).

Kernel	AIMed			BioInfer		
	CV	CL	Δ	CV	CL	Δ
APG	56.2	43.8	**12.4**	60.7	39.1	**21.6**
SL	54.5	42.6	**11.9**	60.0	46.2	**13.8**
kBSPS	44.6	40.3	**4.3**	55.1	47.6	**7.5**
cosine	40.9	37.6	**3.3**	44.1	36.5	**7.5**
edit	39.0	37.0	**2.0**	43.8	31.7	**12.1**
SpT	27.3	28.6	**-1.3**	53.4	43.0	**10.4**

Table 4.1: Performance for cross-validation and cross-learning on AIMed and BioInfer in terms of F_1. Δ represents the difference in F_1 between the two settings (CV-CL). Data from Tikk *et al.* (2010).

4.1.1 Self-training

In this chapter, we explore the usage of self-training for PPI extraction to increase performance on test corpora which have potentially different properties than the training corpus. Note that this essentially includes all current application scenarios in biomedical text mining. Self-training is a special case of semi-supervised learning which, in principle, tries to exploit the large amount of available unlabeled data (Zhu, 2008). It is divided in a number of consecutive steps:

1. Learn a model from manually annotated data.
2. Use this model to label a large pool of unannotated data.
3. Combine the manually and the automatically annotated datasets to create the training data for the final model.

Self-training has been successfully used to improve performance in a number of tasks, including parsing (McClosky *et al.*, 2006b; Reichart and Rappoport, 2007; McClosky and Charniak, 2008), word sense disambiguation (Jimeno-Yepes and Aronson, 2011), and subjectivity classification (Wang *et al.*, 2008). Here we apply self-training to PPI extraction in the following manner. First, we train a model using some gold standard PPI corpora. Second, we apply the model to all sentences from MEDLINE containing at least two protein mentions (excluding those sentences being present in the evaluation corpora). In a second training phase, we augment our original training set by a subset of these classified instances (termed self-trained). Finally, predictive performance of the augmented model is evaluated on previously unseen gold standard corpora.

We compare two strategies for adding self-trained instances and show that we achieve consistent performance improvement across different corpora. The gap in F_1 between CV and CL evaluation is almost halved. We show that self-training is more robust than a fully supervised approach (in terms of gap size standard deviation) making it better suited to assess performance on unlabeled text.

4.2 Methods

For evaluation we use the five benchmark corpora introduced in Subsection 2.5. Since the ultimate goal of PPI extraction is the identification of PPIs in biomedical texts with unknown characteristics, we focus on corpus-wise extrinsic experiments by learning from one or more training corpora. The baseline for our experiments is the CL evaluation scenario, where a classifier is trained on the ensemble of four corpora and evaluated on the fifth. This evaluation is performed exhaustively for all different combinations of training and test sets. We also perform CC evaluations on the two largest corpora, by training a classifier on AIMed and testing on BioInfer and vice versa.

For all experiments we used the shallow linguistic kernel (SL), as it is one of the best performing kernels and produces fairly robust results in extrinsic evaluation (see Figure 4.2 and Table 4.1). In contrast to other methods, SL does not rely on dependency parse trees, whose generation is costly in terms of computation. SVM parameters (*i.e.,* class dependent soft-margin costs C_{+1} and C_{-1}) were set to default values.

4.2.1 Self-training

To augment the training set with automatically created instances we implemented the following workflow (see Figure 4.3): First, we extracted sentence boundaries from MEDLINE citations using the sentence segmentation model of Buyko *et al.* (2006) and scanned these for gene mentions using GNAT (Hakenberg *et al.*, 2011). We found 879,928

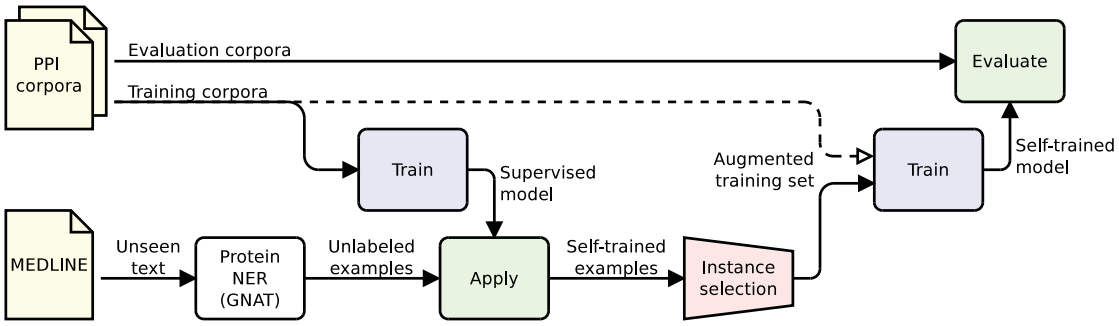


Figure 4.3: Data flow in our self-training setting. The path represented by a dashed line is used in the "self-enriched" strategy but omitted in "self-only."

sentences containing more than one gene mention, totaling in 3,415,624 co-occurrences. For cross-learning, models are trained on the union of four corpora and applied on the unlabeled PPI pairs from MEDLINE. Classified instances are then used to retrain a refined model (termed self-trained model). We evaluated two self-training strategies:

- In the "self-enriched" strategy, we added self-trained instances to the manually annotated training corpora and learned a new model. This setting reflects the common self-training strategy.
- In the "self-only" strategy, we trained solely on the self-trained instances derived from MEDLINE. This setting allows us to investigate the contribution of self-trained instances separate from manually annotated gold standard data.

Self-trained instances are selected by keeping the class ratio equal to the respective training corpus by stratified sampling. This strategy reduces the influence of other parameters and allows us to assess the core contribution of self-training. Note that the class ratio of the evaluation corpus may well be different from that of the (augmented) training set, however, it has been considered unknown at training time to avoid information leakage. The particular instances were selected with respect to their distance to the SVM decision hyperplane, such that the most confidently classified data points were added first. We iteratively increased the amount of self-training examples to a limit of 700,000 training instances. Training on 700,000 instances required 32 GB of main memory and about 32 hours of wall-clock time on a Intel Xeon CPU (X5560 @ 2.80 GHz), while applying the trained model took only 7 msec/instance.

We assessed statistical significance of the results as follows. As advised by Dietterich (1998), we evaluate if one classifier outperforms another classifier by using McNemar's test with continuity correction (McNemar, 1947). The null hypothesis is that both classifiers have the same error rate. Significance of Kendall's correlation coefficients were determined using the Best and Gipps algorithm (Best and Gipps, 1974). The null hypothesis is that the correlation coefficients equal zero. For all tests, we used a

significance level of $\alpha = 0.05$ to determine significance. Neither of these tests makes an assumption on the underlying distribution.

4.3 Results

We show results for the standard (self-enriched) and our custom (self-only) self-training strategies.

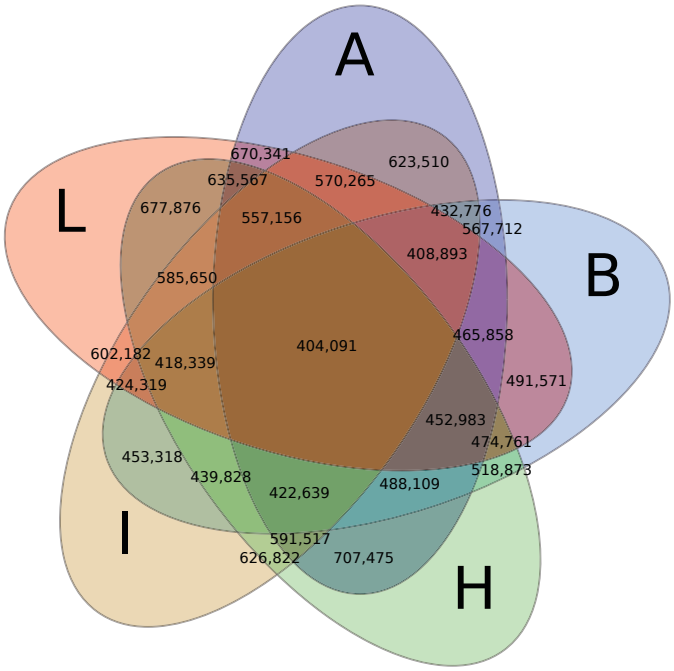
4.3.1 Cross-learning

As shown in Table 4.2, the ratio of positive to negative pairs across the evaluation corpora and the combined training corpora are quite different for most of the five CL experiments. As expected, we observe no correlation (Kendall's tau (τ) = -0.2, p-value = 0.8) for class ratios between evaluation and training corpora. Table 4.2 shows that the number of positive predicted MEDLINE pairs ranges from about 670 k (20 %) to 1,184 k (35 %). The five different models agree on a comparatively large fraction of about 404 k pairs to be positive. Overlaps between different predictions are shown as Venn diagram in Figure 4.4(a) and 4.4(b) for positive and negative instances respectively. Positive to negative ratios between training data and predicted data are correlated but not significant (Kendall's tau (τ) = 0.8, p-value = 0.086). In other words, the classifiers tend to retain class ratios of the original training data when labeling new data. The largest shift in class ratio can be observed for AIMed where positive to negative ratio increases from 0.397 to 0.530. It is important that positive to negative ratios from the evaluation set are not propagated to the training set, as this would approximate the classifier to the evaluation distribution.

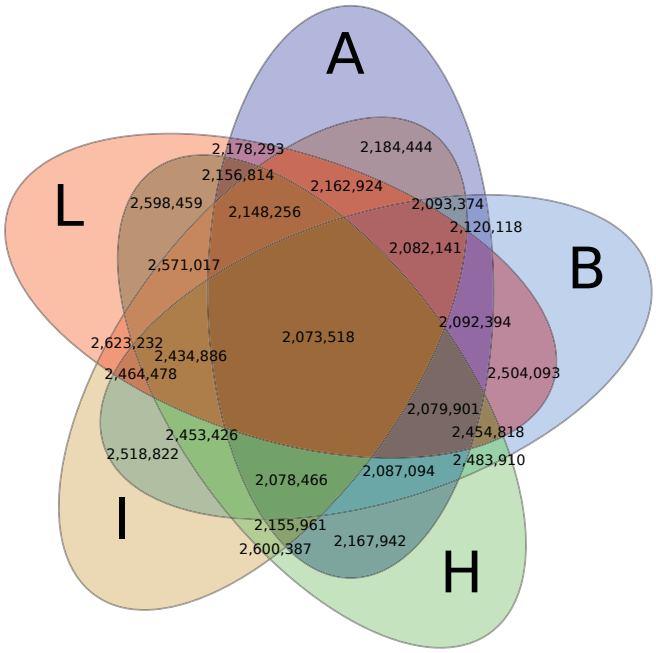
Cross-learning results, for the two self-training strategies with 700,000 instances, are shown in Table 4.3. In comparison to regular cross-learning, F_1 increases considerably for all five corpora in both self-training settings. We observe an average increase in F_1 of 2.6 pp for self-enriched and 3.9 pp for self-only. The average gap between intrinsic cross-validation to extrinsic cross-learning decreases from 9.1 pp to 5.2 pp in F_1. Performance behavior at different fractions of added training instances for AIMed and BioInfer are shown in Figure 4.5.

It is noteworthy that for AIMed, which is the corpus where SL attains lowest precision, self-only leads to an increase of precision of about 1.5 pp. In the remaining four CL settings we observe a small decrease in precision ranging from 1.4 pp to 4.2 pp accompanied by a larger increase in recall from 6.1 pp to 12.2 pp, resulting in consistent improvement in F_1. In addition to F_1, we also evaluate performance in terms of AUC. Analogously to F_1, AUC also improves for both self-training strategies, on average by 0.3 pp for self-enriched and by 0.7 pp for self-only.

McNemar's test assesses the model improvements being significant for the two largest corpora AIMed (p-value $< 1 \cdot 10^{-16}$) and BioInfer (p-value = 0.042). The p-values for the remaining three small corpora range from 0.11 (LLL) to 0.91 (IEPA). It was previously shown by Dietterich (1998) that McNemar's test is very robust, leading to low type I



(a) Overlap for positive instances



(b) Overlap for negative instances

Figure 4.4: Overlap of predictions for the five different classifiers applied on 3,415,624 protein pairs. Classifiers are trained on the union of four corpora excluding the indicated corpus. Single characters (A, B, H, I, L) represent the first letter of the respective evaluation corpus.

Set		AIMed	BioInfer	HPRD50	IEPA	LLL
Evaluation	pos.	1,000	2,534	163	335	164
	neg.	4,834	7,132	270	482	166
	ratio	0.207	0.355	0.604	0.695	0.987
Training	pos.	3,196	1,662	4,033	3,861	4,032
	neg.	8,050	5,752	12,614	12,402	12,718
	ratio	0.397	0.289	0.320	0.311	0.317
Predicted	pos.	1,183,894	679,324	771,263	670,796	723,778
	neg.	2,231,730	2,736,300	2,644,361	2,744,828	2,691,846
	ratio	0.530	0.248	0.292	0.244	0.268

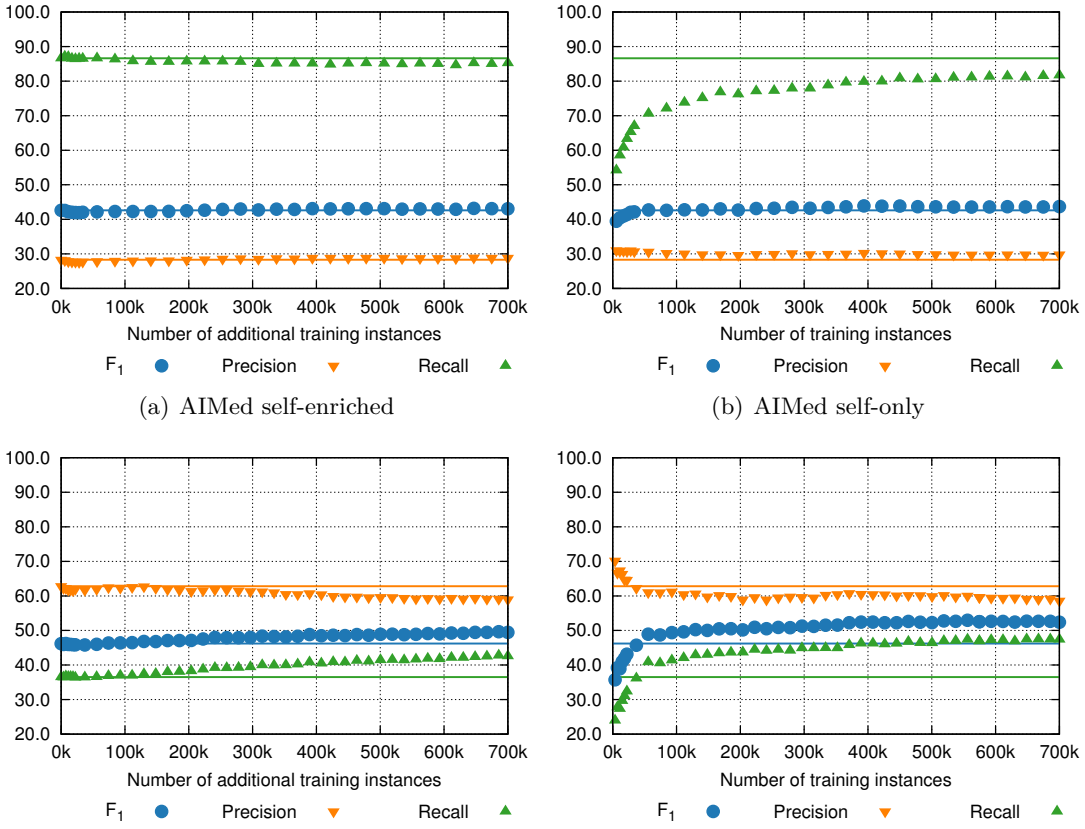
Table 4.2: Comparison of the distribution of positive and negative instances for the different datasets used in the five CL experiments. In each setting one corpus is used for evaluation and the union of the remaining four corpora is used for training. Predicted instances are (originally unlabeled) co-occurring protein pairs taken from MEDLINE annotated by a classifier trained on the corresponding training corpora. These instances are later sampled for self-training.

Method	AIMed			BioInfer			HPRD50			IEPA			LLL			Avg
	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	F_1
CL (baseline)	28.3	**86.6**	42.6	**62.8**	36.5	46.2	**56.9**	68.7	62.2	**71.0**	52.5	60.4	**79.0**	57.3	66.4	55.6
Self-enriched	28.8	85.2	43.1	59.0	42.6	49.5	55.9	72.4	63.1	68.3	61.2	**64.6**	77.0	65.2	70.6	58.2
Self-only	**29.8**	81.7	**43.7**	58.6	**47.5**	**52.4**	55.5	**74.8**	**63.7**	67.2	**61.5**	64.3	77.6	**69.5**	**73.3**	**59.5**
CV	47.5	65.5	54.5	55.1	66.5	60.0	64.4	67.0	64.2	69.5	71.2	69.3	69.0	85.3	74.5	64.7

Table 4.3: CL represents original cross-learning results when training a classifier on the union of four corpora and testing on the fifth. Columns correspond to test corpora. Best results are highlighted in bold. The last column (Avg) covers the macro average F_1 over all five corpora. The row CV provides cross-validation results derived by Tikk *et al.* (2010).

error. Therefore, a possible explanation is that sample sizes (compare with Table 4.2) are to small to observe significant improvements.

Comparing self-enriched with self-only, the self-enriched strategy already performs well with only few induced instances. However, performance increases slower with additional training instances. We attribute this to the behavior of the underlying classification method, *i.e.,* a SVM (see Section 4.4 for detailed discussion).



(a) AIMed self-enriched

(b) AIMed self-only

(c) BioInfer self-enriched

(d) BioInfer self-only

Figure 4.5: Self-training results for AIMed and BioInfer using different quantities of training instances. Horizontal lines represent the baseline CL performance.

4.3.2 Cross-corpus

We also performed cross-corpus experiments for the two largest corpora AIMed and BioInfer. Properties of the two corpora and the prediction step are shown in Table 4.4. Surprisingly, the model trained on AIMed predicts only few positive instances on MEDLINE, whereas the model trained on BioInfer labels more than 50 % as interacting. Self-training results for CC are shown in Table 4.5 and Figure 4.6. Similar to the cross-learning evaluation, both self-training strategies outperform the baseline. Using

the self-only strategy we observe a comparatively large gain of 12.5 pp in F_1 for BioInfer and 3.2 pp in AUC. With 2.1 pp the increase in F_1 is lower but still notable for AIMed. The observed gain in F_1 is again higher for the self-only strategy than using self-enriched. According to McNemar's test model improvements are significant for AIMed (p-value $< 1 \cdot 10^{-32}$) and BioInfer (p-value $< 1 \cdot 10^{-4}$).

Set		AIMed	BioInfer
Evaluation	pos.	1,000	2,534
	neg.	4,834	7,132
	ratio	0.207	0.355
Training	pos.	2,534	1,000
	neg.	7,132	4,834
	ratio	0.355	0.207
Predicted	pos.	1,172,602	295,397
	neg.	2,243,022	3,120,227
	ratio	0.523	0.095

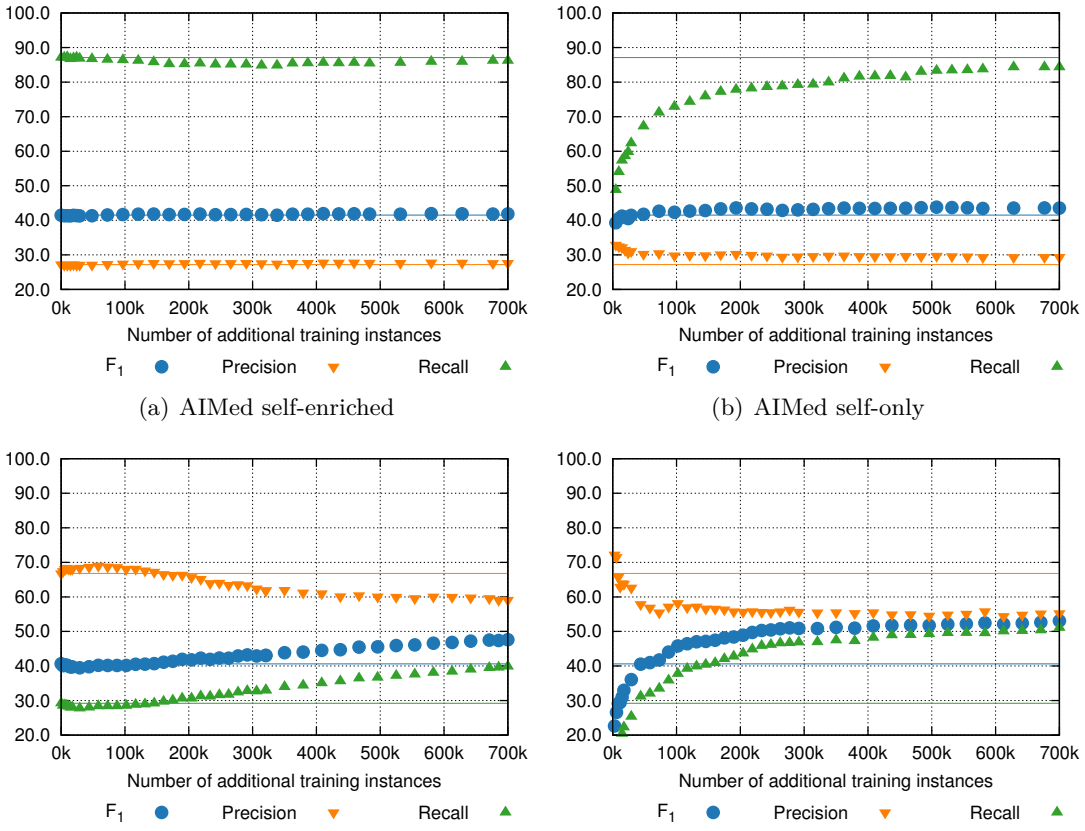
Table 4.4: Comparison of the distribution of positive and negative instances for the two corpora used in CC evaluation.

Method	AIMed			BioInfer		
	P	R	F_1	P	R	F_1
CC (baseline)	27.2	**87.1**	41.5	**66.8**	29.2	40.6
Self-enriched	27.6	86.2	41.9	59.1	39.9	47.6
Self-only	**29.4**	84.3	**43.6**	55.3	**51.1**	**53.1**

Table 4.5: Results for cross-corpus evaluation for AIMed and BioInfer. Classifiers are trained on one corpus and tested on the other one. Columns represent the evaluation corpus.

4.4 Discussion

Both self-training variants performed closer to cross-validation than original cross-learning. Furthermore, both settings reduced the standard error between cross-corpus and cross-learning. For instance, the difference for AIMed between the CL and CC baselines is originally 1.1 pp, whereas self-only CL is about 0.1 pp better than self-only CC (compare Tables 4.3 and 4.5). This performance gap is even higher for BioInfer (5.6 pp), however, applying self-only efficiently reduces the gap to 0.7 pp. Even though the CC self-only model produces slightly better results for BioInfer, we conclude that CL is the more realistic evaluation setting for estimating performance on unseen text. It is simple



(a) AIMed self-enriched

(b) AIMed self-only

(c) BioInfer self-enriched

(d) BioInfer self-only

Figure 4.6: Self-training results for AIMed and BioInfer using different quantities of training instances. Horizontal lines represent the baseline CC performance.

to deduce the best suited training corpus in retrospect, but this information is not available for real tasks. A slight preference for taking the union of all available instances for training was observed by Haddow and Alex (2008), compared to more sophisticated combination techniques. They experimented with combining inconsistent PPI annotations from multiple experts on the same texts to increase in-domain classification performance.

4.4.1 Instance Selection Strategy

Over all five corpora the custom self-only strategy leads to a 1.32 pp higher average F_1 than the well-known self-enriched strategy. We hypothesize its advantage being a result of the instance selection, as we add new data-points with descending order of their distance to the SVM's decision hyperplane maintaining a constant positive/negative ratio. The lesser impact of the self-enriched strategy may be due to the added data-points being less likely to end up as support vectors, however, they do introduce some dimensions in the feature space.

We experimented with another instance selection strategy, by using random sampling instead of adding the most confident (distant) data-points. We observed for small sample sizes an average increase of recall by 10-20 pp and a decrease of precision between 8 pp to 15 pp in comparison to the distant-first method. With additional number of training instances recall increases and precision decreases further and therefore widens the gap between these two measures even more. This effect is also observed for the default self-only strategy, however precision is higher and decreases less with additional number of examples. We would therefore not recommend random sampling.

We also evaluated a fourth instance selection strategy, by adding data-points closest (instead of distant) to the SVM decision hyperplane. This means that we add data points with uncertain prediction scores first and subsequently add more confident data points. For small sample sizes, the closest-first method leads to an almost random classifier. However, the classifier gained in precision and recall with increasing number of training instances, that is, with the addition of more distant data-points. Unfortunately, training time substantially increases using this selection strategy. For example, training with 200,000 instances lasts more than a week. Furthermore, the number of support vectors increases to approximately 150,000, which is about 17 times higher in comparison to self-only and self-enriched, using the most distant data points. We conclude that data-points close to the decision hyperplane should be excluded from training.

4.5 Conclusion

We have shown that self-training, a semi-supervised technique, can be used to consistently improve extrinsic performance. This is the most realistic setting when assessing performance of a classifier for a corpus, such as MEDLINE, where the underlying characteristics are hidden and only partially covered by any corpus. On five evaluation corpora we achieved an average improvement of 3.9 pp F_1 (ranging from 1.1 pp to 6.9 pp) over a well performing baseline. Taking all results into account, we conclude that self-training is beneficial when applying a model on a text corpus with unknown characteristics.

One disadvantage of self-training is the high demand of computational time and main memory to train a classifier. It could be worthwhile to integrate feature selection techniques into the relationship extraction classifier to identify and remove irrelevant features. It has been previously shown that this can positively effect training time and performance for event extraction (Landeghem *et al.*, 2010).

There are several ways to expand upon our work. It would be interesting to investigate for which type of sentences self-training is most useful. In similar studies for parsing, McClosky *et al.* (2006a) showed that self-training is most beneficial for medium-ranged sentences, while Reichart and Rappoport (2007) showed that the number of previously unseen words in a sentence is an indicator of benefit for a self-trained model. This analysis would allow for selecting the optimal model for each sentence according to its syntactic properties.

Future work will have to examine the question of how much additional self-trained instances are needed to build a better classifier. Our results indicate that more instances are generally advantageous and at least not harmful as F_1 usually converges. However, a deeper investigation is required for a better understanding of convergence properties, useful to determine a proper stopping criteria based on unlabeled data.

4.6 Related Work

This section describes the most closely related publications involving domain adaptation or semi-supervised learning. To the best of our knowledge, no other publication investigated strategies to improve extrinsic performance for biomedical relationship extraction.

Miwa *et al.* (2009a) (see Subsection 2.5.1) use corpus weighting, another domain adaptation technique, to utilize annotations distributed in different PPI corpora. The approach distinguishes between in-domain data (target) and out-domain data (source). Corpus weighting extends the original soft-margin SVM problem by incorporating different cost parameters for source and domain data (see Formula 2.16). This strategy allows to increase F_1 on small corpora (*i.e.,* HPRD50, IEPA, and LLL) but provides only little or no improvement on AIMed and BioInfer. Most importantly, corpus weighting assumes that the target domain is known during training by implicitly defining the target corpus. This significantly differs from our approach, where the target domain is hidden from the learning algorithm (CL and CC evaluation). For these reasons, corpus weighting is not comparable to our approach, as it was only evaluated in intrinsic experiments.

Erkan *et al.* (2007) use transductive learning for PPI extraction. The goal of transductive learning is to include the evaluation data during training. Class labels of the substituted test instances are removed during training in order to avoid over-fitting of the classifier. This setting differs from typical supervised machine learning by knowing the evaluation instance in advance. In other words the classifier does not need to learn a general model, but rather a model performing well on the unlabeled test instances. As an analogy, we can think about an exam where students know the test questions (without the answers) in advance. The usual (machine learning) goal is to generalize

from the training data in order to perform well on arbitrary questions. Transductive support vector machines (TSVMs) reformulate the original soft margin SVM formulation by searching for the max margin hyperplane separating training and unlabeled test data. TSVM poses a non-convex optimization problem, thus requiring constraints to search for an approximate solution. For example, a constraint introduced by Joachims (1999) is that the class ratio is equal between training and test sets. Such an assumption would be clearly violated in an extrinsic evaluation setting (see Table 4.2). Again, no direct comparison of the obtained results can be made due to the fundamental differences in the experimental settings (CV versus CL). Furthermore, training time with TSVM increases enormously compared to non-transductive SVM (Tikk *et al.*, 2010), rendering its application in a realistic setting (with hundreds of thousands of examples) impossible.

Björne *et al.* (2012) tested the effect of self-training for event extraction in the context of the BioNLP'11 shared task. MEDLINE wide event type predictions are collected from the EVEX database (Van Landeghem *et al.*, 2011) and prediction values are normalized to the standard normal distribution (*i.e.,* with a mean of 0 and a variance of 1). Additional instances are then randomly sampled from different confidence intervals. Results indicate that high confidence intervals provide the most benefit on the development set with an increase of up to 1.4 pp in F_1. The effect on the official test data remains with 0.4 pp F_1 rather small. In difference to Björne *et al.* (2012) we evaluated the impact of self-training in an extrinsic setting in order to improve robustness, whereas Björne *et al.* applied self-training to improve in-domain performance.

MacKinlay *et al.* (2013) follow a co-training inspired procedure for the extraction of biomedical events. The authors learn patterns (dependency subgraphs) on the manually annotated BioNLP'13 shared task training data. Additional training instances are generated by applying the event extraction system TEES on MEDLINE and PMC. The most confidently identified events of TEES are used to complement the manually annotated training data by extracting additional patterns. This strategy leads to a substantial increase of approximately 3.5 pp in F_1 on the development set, but had only minor effect (0.4 pp) on the test set.

5 Distant Supervision

Previous chapters covered the foundations of binary relationship extraction. Supervised relationship extraction approaches, as covered in this thesis, learn a model from manually annotated data. However, manual annotation is time consuming and often biased to the annotation guideline and corpus selection criterion. To overcome this issue, recent work has introduced the concept of distant supervision (Craven and Kumlien, 1999; Mintz *et al.*, 2009). Instead of manually annotated corpora, distant supervision infers training instances from non-annotated texts using knowledge bases. This allows to increase training set size by some orders of magnitude in comparison to manual annotation. However, corpora derived by distant supervision are inherently noisy, thus benefiting from robust relationship extraction methods.

In this chapter we analyze the usability of distant supervision for protein-protein interaction extraction using two different learning approaches. The first approach uses SVM as statistical learner[1], whereas the second approach learns graphical patterns from the dependency tree[2].

5.1 Introduction

Distant supervision is a semi-supervised learning technique often used in the context of relationship extraction from text. The method has been originally presented by Craven and Kumlien (1999), while the term distant supervision has been coined by Mintz *et al.* (2009). The idea of distant supervision is to automatically generate training data without manual intervention. The general distant supervision approach for relationship extraction is depicted in Figure 5.1 and works as follows:

① Identify a knowledge base that contains pairs of entities about the relationship-type in question (*e.g.,* PPI-database).

② Compile a large unannotated text resource relevant for the target domain (*e.g.,* MEDLINE abstracts).

③ Recognize and normalize relevant named entities (*e.g.,* protein names).

④ Associate entity-pairs from the knowledge base with previously identified instances in the text corpus. Entity pairs contained in the knowledge base are labeled as positive instances. Negative instances are labeled by following the *closed world assumption*. The closed world assumption states that entity pairs lacking in the knowledge base do not feature the relationship type in question.

[1] Joint work with I. Solt, T. Bobic, R. Klinger, and U. Leser
[2] Joint work with S. Pietschmann, I. Solt, D. Tikk, and U. Leser

⑤ Learn a classifier on the distantly labeled corpus.

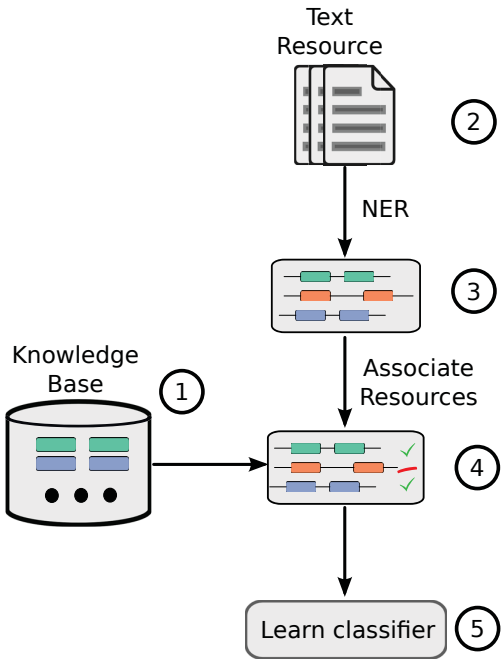


Figure 5.1: Distant supervision workflow.

5.1.1 Problems of Distant Supervision

Corpora annotated by the distant supervision and closed world assumption contain a certain amount of mislabeled examples. First, some entity pairs do not describe the desired relationship and lead to false positives. For instance, the sentence "c-Myc and BRCA1 are both human proteins." provides no evidence for a protein-protein interaction between the named entities c-Myc and BRCA1. The second type of error concerns incomplete knowledge bases, leading to false negative annotations. If the knowledge base misses information about a relationship between an entity pair, distant supervision will wrongly label these instances as negative. This problem is especially severe for transient relationships, such as "person lives in city" or "corporate ownership". But this problem also occurs for all types of relationships where the knowledge bases are incomplete (*e.g.,* PPI-databases).

For these reasons distant supervision requires robust relationship extraction methods capable of dealing with both types of noise. In this chapter, we will discuss two different methods for extracting PPI from biomedical texts using distant supervision. One method relies on support vector machines, whereas the other learns patterns from the dependency graph. Both methods incorporate different ideas to remove the noise in distantly labeled training data.

Testing the distant supervision assumption

Before applying the afore mentioned method for building a distantly labeled corpus we evaluated the potential of distant supervision in the context of PPI extraction using the following methodology: First, we identified all gene/protein occurrences in the two largest PPI corpora (AIMed and BioInfer) using GNAT. Second, we used Intact to label identified protein pairs using the distant supervision approach. Gene name recognition achieves a remarkable precision of 91.6 % and 96.3 % on AIMed and BioInfer, respectively (exact match). However, recall values are rather low with 32.2 % for AIMed and 22.1 % for BioInfer. Normalized gene-IDs are not provided for these corpora and can therefore not be evaluated.

We then evaluated the overlap between the distantly labeled corpora and the original manually annotated data. In other words we evaluated the agreement between manual annotation and distant supervision. The result of this evaluation is shown in the confusion Tables 5.1 and 5.2 for AIMed and BioInfer. Significance of association has been determined using χ^2-test where the null hypothesis states that the two data sources are independent of each other. The result proved to be significant for both corpora with p-values of $4.6 \cdot 10^{-6}$ for AIMed and 0.003 for BioInfer and indicates that information derived by distant supervision overlaps with manually annotated corpora.

		Manual annotation	
		positive	negative
Distant supervision	positive	42	74
	negative	103	512

Table 5.1: Confusion table between corpus annotation and distant supervision for AIMed.

		Manual annotation	
		positive	negative
Distant supervision	positive	58	216
	negative	70	457

Table 5.2: Confusion table between corpus annotation and distant supervision for BioInfer.

In another experiment we used the data published by Thomas *et al.* (2012b). In this experiment Thomas *et al.* labeled approximately 50,000 MEDLINE abstracts using the SL classifier trained on a distantly labeled corpus. The authors then trained the same relationship extraction extraction algorithm using the union of all five manually annotated PPI corpora. The inter-classification agreement between these two classifiers is 86.4 % and is highly significant according to a χ^2-test (p-value $< 2.2 \cdot 10^{-16}$). In other

words the classifier trained on five manually annotated corpora and the same classifier trained on a distantly labeled corpus agree on 86.4 % of all 50,000 predictions.

5.2 Using Support-Vector Machines

This section analyzes the ability to learn SVM models from distantly labeled text. Trained models will be analyzed on manually annotated corpora.

5.2.1 Training Data Generation

Distantly labeled training instances are generated as follows: All MEDLINE citations published between 1985 and 2011 are split into sentences using the sentence segmentation model from Buyko *et al.* (2006) and scanned for gene and protein names using GNAT (Hakenberg *et al.*, 2011). In total, we find 1,312,059 sentences containing 8,324,763 protein pairs. To avoid information leakage between training and test sets, articles contained in any of the five evaluation corpora have been excluded from the distantly labeled corpus. This procedure excludes 7,476 ($< 0.1\%$) protein pairs from the training set. Following the distant supervision approach protein pairs that are contained in the PPI knowledge base IntAct[3] (Aranda *et al.*, 2010) are labeled as positive instances. Co-occuring protein pairs not contained in IntAct are labeled as negative instances.

As argued in Subsection 5.1.1 it is likely that negative and positive instances contain a certain amount of mislabeled examples (*i.e.,* false positives, false negatives). To counteract this effect we utilize different heuristics minimizing the amount of mislabeled instances. Firstly, we generate a list of words, which are frequently employed to indicate a protein-protein interaction[4]. This list is used to filter positive and negative instances such that positive instances must contain at least one interaction word (*pos-iword*) and negative must not contain an interaction word (*neg-iword*). Application of both filters in combination is referred to as *pos/neg-iword*. Secondly, we assume that sentences mentioning only two proteins are more likely to describe a relationship between these two proteins than sentences containing several protein names. This filter is called *pos-pair*. For the sake of completeness, it is tested on negative instances alone (*neg-pair*) and on positive and negative instances in combination (*pos/neg-pair*). All seven settings are summarized in Table 5.3.

5.2.2 Classification

As classification algorithm we use a Support-Vector machine employing the shallow linguistic (SL) kernel, which achieves state-of-the-art performance and requires no parse-tree information. This allows us to collect large training data without the need of time consuming parse-tree generation.

Using all distantly labeled instances during training proved as too time expensive. Classifiers are therefore trained with a small subset from all 8,3 million pairs, using

[3] As of March 24, 2010.

[4] `http://www2.informatik.hu-berlin.de/~thomas/pub/iwords.txt`

Setting	Feature:	Interaction word count		Pairs in sentence	
	Condition:	≥ 1	$= 0$	$= 1$	$= 1$
	Applied to:	positive	negative	positive	negative
baseline					
pos-iword		•			
neg-iword			•		
pos/neg-iword		•	•		
pos-pair				•	
neg-pair					•
pos/neg-pair				•	•

Table 5.3: All seven experimental settings. Based on the number of interaction words and protein mention pairs in the containing sentence, we filter out automatically generated positive or negative example pairs not meeting the indicated heuristic condition. The dots indicate which filter is applied for which setting. For instance no filtering takes place for the baseline setting.

50,000 instances in all experiments except when stated differently. We also evaluate how much training data is required to successfully train a classifier and if the classifier reaches a steady performance after a certain number of training instances.

Another well acknowledged problem is that classifiers often tend to keep the same positive to negative ratio seen in the training phase (Chawla *et al.*, 2004). This raises the question on how training class distribution should be set. In our first experiments, we set the class ratio according to the class distribution averaged over all corpora excluding the evaluation corpus. This allows us to directly compare classifier performance with cross-learning results using the same classifier (see also Chapter 4). Influence of class imbalance is evaluated separately by varying positive to negative ratios from 0.001 to $1,000$ using the best filtering strategy from the previous experiment.

Negative instances are generated using the *closed world assumption*, stating that two co-occurring protein mentions are annotated as negative when not contained in the knowledge base. To estimate the impact of the closed world assumption, we experimented with another technique by using the Negatome database[5] (Smialowski *et al.*, 2010) to infer negative examples. Negatome provides a reference set of *non-interacting* protein pairs and is thus better suited to infer negative examples than the closed world assumption. Unfortunately, reliable information about non-interaction is difficult to obtain and therefore the database contains far less entries than IntAct. From our 8 million co-occurring protein pairs only 6,005 could be labeled as certainly negative by Negatome. This is insufficient to build a reasonable set of negative training instances for our experiments. Additional negative training instances required for training are therefore inferred using the closed world assumption.

[5] As of April 30, 2011.

Finally, we evaluate if a majority voting ensemble of 11 classifiers trained on randomly drawn training instances can further improve extraction quality. This approach loosely follows a bagging strategy (Breiman, 1996, see also Section 3.1). However, training instances are less overlapping than using the regular bagging strategy. Bagging generates new training sets by sampling instances from the original dataset with replacement. In difference to this, we sample instances from the original dataset without replacement. The latter strategy can be applied due to the huge number of available training instances.

5.2.3 Evaluation

For evaluation, we use the five PPI corpora AIMed, BioInfer, HPRD50, IEPA, and LLL introduced in Subsection 2.5. Each experiment is repeated 10 times with randomly sampled training instances. This strategy results in 10 independent estimates for precision, recall, F_1, and AUC and allows to robustly estimate individual evaluation metrics. p-values between different experiments are derived using single sided Mann–Whitney U test (Mann and Whitney, 1947), with the null hypothesis that the median of two samples is equal. Significance of Kendall correlation (Kendall, 1938) is determined using Best and Gipps (1974) with the null hypothesis that correlation equals zero. For all tests we use a significance level of $\alpha = 0.01$.

5.2.4 Results

Mean values for the seven different instance selection strategies (see Table 5.3) are displayed in Table 5.4. All strategies, except *neg-pair* filtering, obtain an AUC higher than 0.5. The difference in AUC is generally significantly better than 0.5, except for three experiments using the smallest corpus (LLL). AUC is identical to the probability that a classifier ranks a randomly chosen negative instance lower than a randomly chosen positive instance. Therefore, AUC scores above 0.5 show that the distant supervision assumption holds, to at least some extend, for PPI extraction.

The various settings introduced to filter out likely noisy training instances either improved precision or recall or both over the baseline of using automatically labeled instances without applying any filters. Many instance selection strategies for AIMed, BioInfer and HPRD50 significantly outperform co-occurrence in terms of F_1. However, co-occurrence significantly outperforms all seven settings for the two remaining corpora IEPA and LLL in F_1. This might have several reasons: First, these two corpora have the highest fraction of positive instances, therefore co-occurrence is a stronger baseline. Second, IEPA describes chemical relations instead of PPIs, thus our training corpus might not properly reflect the syntactic property of such relations.

It is encouraging that on two corpora (BioInfer and HPRD50) the best setting performs about on par with the best cross-learning results from Tikk *et al.* (2010), which have been generated using manually annotated data and are therefore suspected to produce superior results. Distant supervision on the other hand labels text corpora without human intervention, thus reducing the cost of generating training corpora.

For each corpus we calculate the average rank in F_1 for the seven different instance

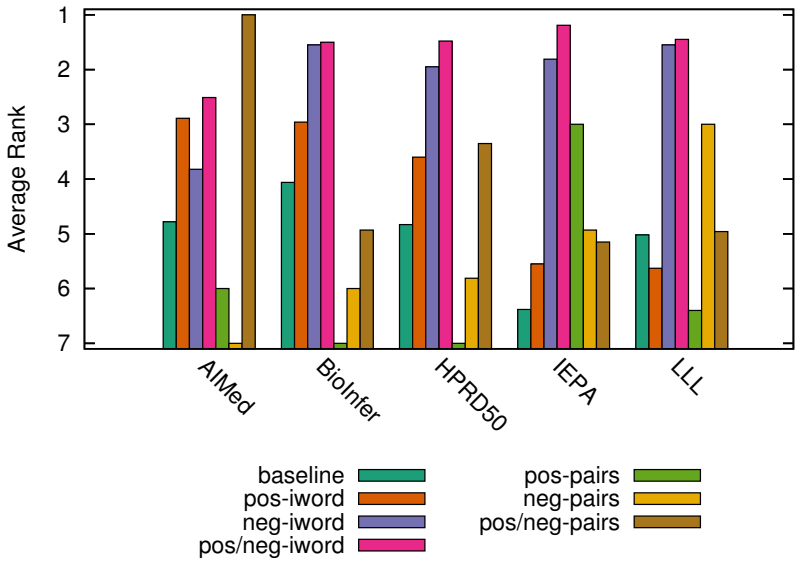


Figure 5.2: Average rank in F_1 for each experimental setting on the five evaluation corpora.

filtering strategies (see Figure 5.2). Figure 5.3 shows how often a selection strategy significantly supersedes the remaining six strategies in terms of F_1 (according to Mann–Whitney U test). For instance, pos/neg-iword significantly outperforms all other six strategies on the IPEA corpus. The same strategy outperforms only four other strategies on AIMed. Figure 5.2 and Figure 5.3 indicate that the filters pos/neg-iword and neg-iword perform well across all five corpora, suggesting superior robustness for these two settings. These strategies are only once outperformed across all five corpora: On AIMed the filtering strategy pos/neg-pairs significantly outperforms all other strategies. However, it achieves mediocre results on the remaining four corpora which indicates a comparably lower robustness. This filtering strategy would therefore not be advised as it provides superior results on only one corpus. In the following, we analyze and compare different instance selection strategies in more detail.

Interaction word based settings

All experiments using our interaction words for instance selection lead to an increase of F_1 and AUC. In comparison to distant supervision without filtering (baseline) we observe the highest increase in AUC (3.8 pp) as well as F_1 (11.8 pp) using filtering of positive and negative instances together (pos/neg-iword). This strategy is closely followed by filtering only negative instances (neg-iword) with an average F_1 improvement of 11.3 pp. Finally we observe only a marginal improvement of 1.3 pp in F_1 when exclusively filtering positive instances (pos-iword).

Method		AIMed				BioInfer				HPRD50				IEPA				LLL			
		AUC	P	R	F_1	AUC	P	R	F_1	AUC	P	R	F_1	AUC	P	R	F_1	AUC	P	R	F_1
co-occurrence			17.8	(100)	30.1		26.6	(100)	41.7		38.9	(100)	55.4		40.8	(100)	57.6		55.9	(100)	70.3
cross-learning (Tikk *et al.,)*		77.5	28.3	86.6	42.6	74.9	62.8	36.5	46.2	78.0	56.9	68.7	62.2	75.6	71.0	52.5	60.4	79.5	79.0	57.3	66.4
Setting	baseline	65.1	21.0	82.8	33.5	63.2	33.3	64.2	43.8	64.4	42.8	75.4	54.6	52.2	40.9	11.6	18.0	51.8	51.3	39.2	44.4
	pos-iword	66.6	21.8	82.6	34.5	67.5	38.4	60.8	47.1	67.5	45.5	76.5	57.1	53.8	48.6	12.3	19.6	51.6	50.0	37.0	42.2
	neg-iword	65.3	21.1	**91.1**	34.2	68.1	37.3	70.9	48.9	**73.4**	43.9	**93.6**	59.8	54.7	43.9	49.9	46.7	53.9	49.9	**77.4**	60.7
	pos/neg-iword	65.1	21.4	89.8	34.6	68.6	38.6	67.0	**49.0**	73.3	44.8	93.2	**60.5**	54.6	43.8	**53.2**	**48.0**	53.5	50.7	75.8	**60.8**
	pos-pairs	64.2	**29.3**	33.4	31.2	**69.8**	**57.8**	18.0	27.5	62.7	**47.9**	35.6	40.8	**66.6**	**54.9**	26.3	35.5	**63.2**	**68.2**	27.8	39.5
	neg-pairs	46.9	17.2	85.5	28.6	37.3	24.4	**85.6**	37.9	50.8	39.0	80.9	52.6	36.5	22.4	18.6	20.3	38.2	44.7	66.2	53.3
	pos/neg-pairs	**69.7**	23.6	82.3	**36.6**	62.0	32.8	60.6	42.5	69.2	46.5	75.2	57.5	56.0	43.4	13.3	20.3	54.3	54.5	37.9	44.6
Setting (+Negatome)	baseline	65.9	22.2	79.6	34.7	65.7	36.8	58.6	45.2	67.6	46.7	74.0	57.3	54.9	47.5	12.7	20.0	54.8	53.6	36.3	43.2
	pos-iword	67.4	22.9	81.4	35.8	69.1	41.1	56.3	47.5	69.2	**47.9**	75.4	58.5	57.4	52.6	12.9	20.6	52.3	51.2	37.5	43.1
	neg-iword	65.3	21.1	**90.7**	34.3	68.8	38.1	69.6	**49.2**	**73.6**	44.6	92.1	60.1	55.6	44.4	51.7	47.8	55.2	51.3	**78.9**	62.2
	pos/neg-iword	65.1	21.4	89.4	34.6	68.8	38.8	66.9	49.1	73.2	44.8	**92.2**	**60.3**	55.3	44.2	**53.8**	**48.5**	54.9	52.2	77.9	**62.5**
	pos-pairs	64.6	**29.6**	33.7	31.5	**69.7**	**58.2**	18.3	27.8	62.2	48.5	35.5	41.0	**66.9**	**56.6**	30.7	39.7	**63.4**	**68.8**	28.1	39.9
	neg-pairs	47.0	17.2	84.9	28.6	37.0	24.3	**85.0**	37.8	50.9	38.4	79.8	51.9	36.0	22.4	18.5	20.3	38.5	45.1	66.0	53.5
	pos/neg-pairs	**69.8**	23.8	81.1	**36.8**	63.9	34.6	58.6	43.5	69.5	47.5	74.2	57.9	57.0	44.3	13.9	21.1	54.7	53.2	34.5	41.7
Train pos/neg ratio	1,000	60.6	19.0	89.8	31.3	64.2	31.3	**84.6**	45.7	62.5	41.1	92.9	57.0	57.9	42.6	**88.3**	**57.3**	61.2	53.7	**93.3**	**68.1**
	100	63.9	20.0	88.7	32.7	69.0	35.5	77.8	48.7	71.5	44.2	91.9	59.6	**58.9**	45.6	65.6	53.7	**61.5**	53.1	85.8	65.6
	10	65.5	20.9	91.0	33.9	**71.2**	38.7	76.0	**51.2**	74.1	44.2	**95.8**	60.5	57.9	45.7	55.5	50.1	57.9	51.8	80.7	63.1
	1	65.6	21.4	**91.1**	34.7	70.0	38.6	71.3	50.1	**74.5**	44.3	95.5	**60.6**	56.1	45.0	55.5	49.7	55.7	51.6	79.3	62.5
	0.1	65.4	22.3	81.3	**35.0**	67.9	40.9	57.9	48.0	72.1	46.9	84.7	60.4	53.5	43.1	37.9	40.3	51.0	50.0	58.7	53.9
	0.01	**66.0**	26.9	46.7	34.1	66.5	46.9	24.7	32.4	70.4	59.7	48.5	53.4	52.8	**48.2**	8.3	14.2	52.2	**54.3**	12.3	19.7
	0.001	61.5	**41.4**	0.9	1.8	63.2	**63.0**	0.3	0.6	67.8	**72.5**	1.3	2.6	53.0	30.0	0.1	0.2	54.1	10.0	0.1	0.1
Train set size (total)	500	63.4	**21.8**	71.5	33.4	65.9	39.8	44.6	41.9	67.6	**48.4**	67.4	56.2	55.5	45.4	31.5	36.7	54.0	**52.6**	53.4	52.7
	5,000	65.3	21.4	84.3	34.2	69.0	**39.9**	63.5	48.9	72.6	45.7	89.0	60.4	**56.8**	**46.1**	**41.9**	43.8	54.5	51.3	66.3	57.8
	15,000	**65.5**	**21.6**	87.9	**34.6**	**69.1**	39.7	65.1	**49.3**	**74.2**	45.6	92.9	**61.2**	55.8	44.5	47.4	45.9	**55.7**	51.9	75.1	61.3
	30,000	65.3	21.5	89.4	**34.6**	68.8	39.2	66.5	**49.3**	73.0	44.6	**93.1**	60.3	55.0	44.0	50.7	47.1	53.8	50.9	75.2	60.7
	70,000	65.1	21.3	90.7	**34.6**	68.6	38.1	67.4	48.7	73.2	44.2	92.1	59.8	54.2	43.7	55.0	48.7	53.9	50.9	78.6	61.8
	150,000	64.7	21.3	**91.3**	34.5	68.2	37.5	**68.1**	48.4	73.1	44.1	92.8	59.8	53.0	43.0	57.1	**49.1**	52.7	51.1	**81.3**	**62.7**

Table 5.4: Results of different instance selection strategies, employing Negatome as negative knowledge base, different positive to negative ratios in the training set, and total sample size.

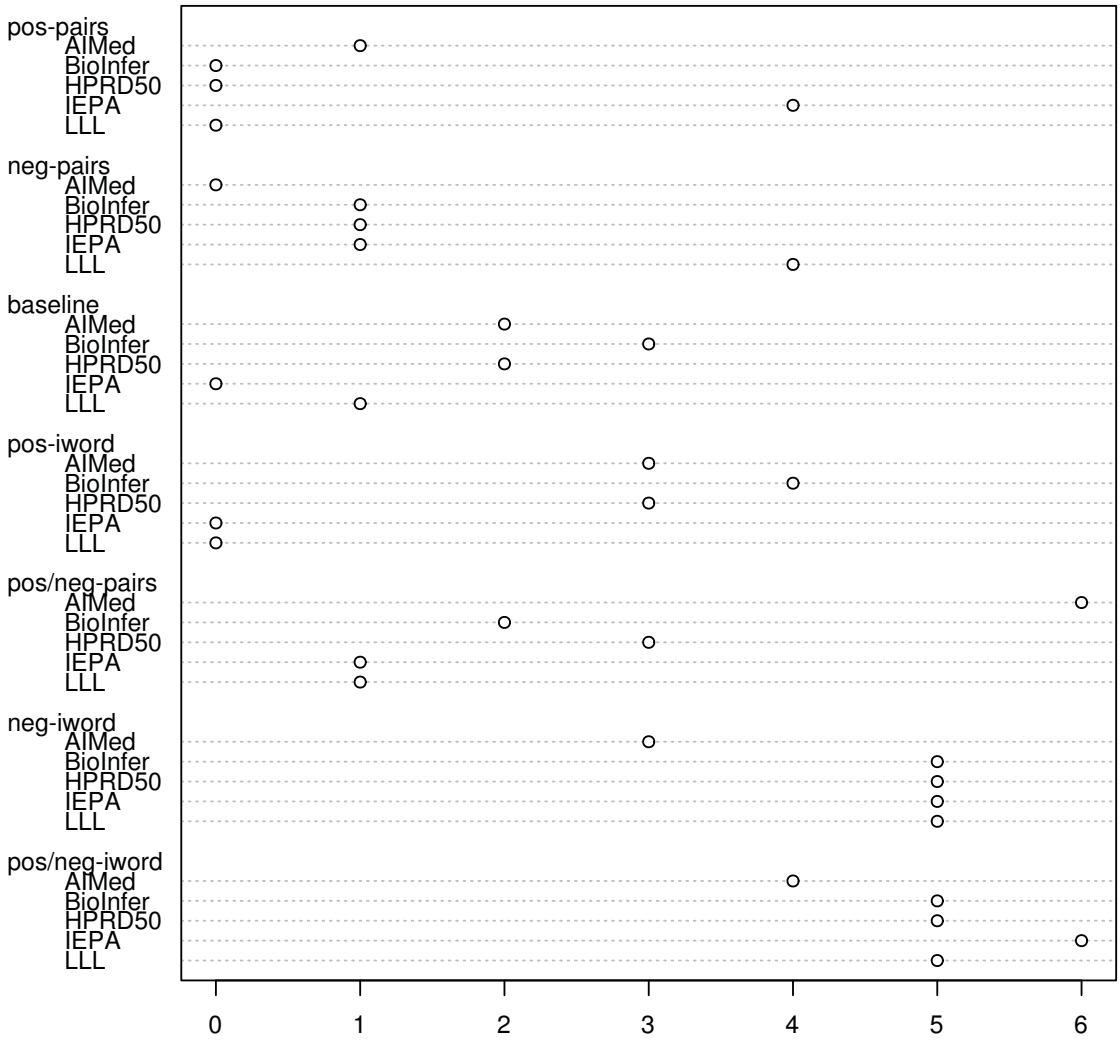


Figure 5.3: Comparison of all seven instance selection strategies. Individual points represent how often a specific instance selection strategy significantly outperforms the remaining six strategies for a given corpus. For instance, pos/neg-iword significantly outperforms all other six strategies on the IPEA corpus, but only four strategies on AIMed. Strategies (*i.e.,* pos-pair, neg-pair, baseline, ...) are ranked by the total number of times the strategy significantly superseded others across all corpora.

Negatome

We repeated the previously introduced instance filtering techniques, but inferred negative training instances by using the Negatome knowledge base. From all 8,324,763 co-occurring protein pairs found in MEDLINE, Negatome allows us to label 6,005 as negative. To account for the relatively small number of negative instances, additional instances are drawn from the set of instances derived by the closed world assumption. Using Negatome leads to a small increase of 0.5 percentage points in F_1, due to an average increase of 1.1 pp in precision over all five corpora and seven settings. We also observe a tendency for increased AUC (0.9 pp). The largest gain in precision (3.5 pp) is observed between the two baseline results when no instance filtering is applied. Altogether the experiments with Negatome indicate that knowledge bases for non-interacting protein pairs provide a better source to infer negative instances than the closed world assumption.

A clear drawback of Negatome is the comparable small size. On our dataset we could only generate 6,005 negative instances using Negatome. The number of negative training instances could be increased by generalizing proteins across species using information about homologous genes (*e.g.,* using the Homologene database). Using this approach on our data set we could infer approximately 4,200 additional training instances. However, it is unclear if these derived instances are of the same quality than the original 6,005 negative instances.

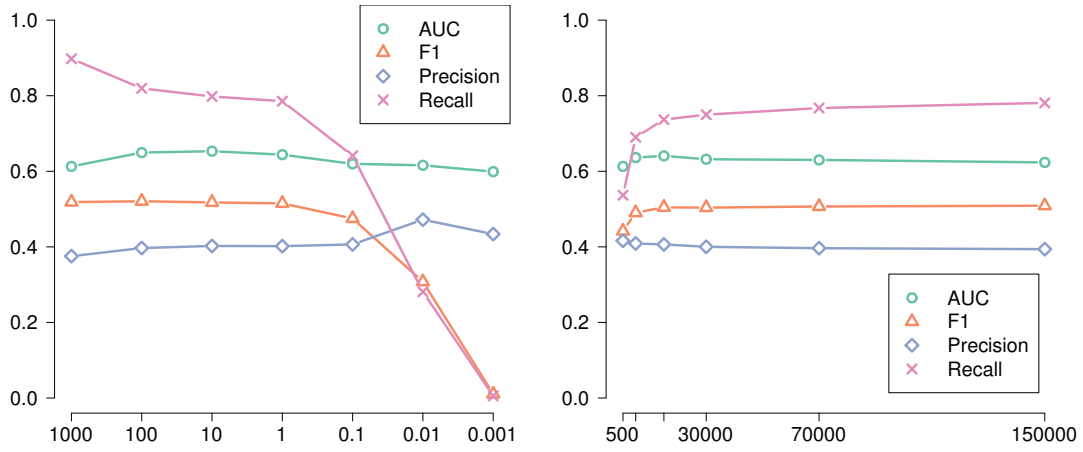
Effect of the positive to negative ratio

Results for varied positive to negative training ratios are shown in Figure 5.4(a) and Table 5.4 (see Page 86). As expected, the results show that positive to negative ratio on training data affects performance of a classifier. Precision and recall strongly correlate with the pos/neg ratio seen in the training set. The strong correlation between recall and pos/neg ratio (Kendall's tau ranging from 0.524 to 1 for all five corpora) is expected, as the classifier tends to assign more test instances to the majority class. Oversampling of positive training instances works best for corpora with high fractions of positive examples. A strong correlation (Kendall's tau ranging from -0.9 to -1.0) between precision and class ratio can be observed for AIMed, BioInfer, and HPRD50. Correlation for IEPA is close to zero and for LLL the correlation is even positive but not significant (p-value of 0.13). Overall, the observed influence of class imbalance is less pronounced than expected. For instance F_1 remains comparably robust with an average standard deviation of 2.6 pp for ratios between 0.1 and 10, whereas in a range between 1 and 100 the average standard deviation increases to about 11 pp. With more pronounced differences in the training ratio, a strong impact on F_1 can be observed.

Effect of training set size

The impact of training set size on different corpora is shown in Table 5.4 (see Page 86). Results aggregated over all corpora are shown in Figure 5.4(b). With increasing training set size a monotonic increase in recall (Kendall's Tau of 1; p-value < 0.01) can be

observed for all corpora, except HPRD50. The negative correlation between precision and sample size is less pronounced but still observable for all five corpora as Kendall's Tau ranges between -0.552 and -1. For this reason, F_1 increases for corpora with many positive instances.



(a) Positive to negative ratio

(b) Sample size

Figure 5.4: Distribution of mean precision, recall, F_1, and AUC aggregated over all five corpora for different class ratios and training set sizes.

Bagging

Based on the previous experiments we determined the best individual strategies: Filtering of positive and negative instances for interaction words, a positive to negative class ratio of 1, and a training size of 15,000 instances. We sampled 11 individual training sets exhibiting these properties from the population of all distantly labeled MEDLINE instances. For each training set we learned a SVM employing the SL kernel. The minimum, average, and maximal performance of all individual classifiers are shown in Table 5.5, together with the results for majority voting (bagging).

Bagging performs about on par with the mean of the individual classifiers and we observe, according to Mann-Whitney U-test, no significant difference between bagging and the 11 classifiers. However, single classifier sometimes performs better or worse, whereas bagging always performs close to the average. Individual classifier performance can deviate between 0.4 pp on AIMed to 4.0 pp on IEPA. Thus, bagging can be successfully applied for improving robustness of a classifier.

5.2.5 Conclusion

We investigated the use of distant supervision combined with a machine learning approach to detect protein-protein interactions. We demonstrated that distant supervision

Method	AIMed				BioInfer				HPRD50				IEPA				LLL			
	AUC	P	R	F_1	AUC	P	R	F_1	AUC	P	R	F_1	AUC	P	R	F_1	AUC	P	R	F_1
co-occurrence		17.8	(100)	30.1		26.6	(100)	41.7		38.9	(100)	55.4		40.8	(100)	57.6		55.9	(100)	70.3
cross-learning (Tikk *et al.*,)	77.5	28.3	86.6	42.6	74.9	62.8	36.5	46.2	78.0	56.9	68.7	62.2	75.6	71.0	52.5	60.4	79.5	79.0	57.3	66.4
min of 11 runs	64.7	21.1	90.3	34.3	69.2	68.7	38.2	49.8	73.0	43.4	93.2	59.4	54.3	43.3	51.0	46.9	53.8	49.2	75.6	59.8
mean of 11 runs	65.5	21.4	90.9	34.6	69.9	70.7	38.9	50.2	74.0	44.4	94.7	60.4	55.5	44.7	54.6	49.1	55.2	50.6	78.0	61.4
bagging over 11 runs		21.4	91.3	34.7		70.9	39.3	50.6		44.3	95.1	60.4		44.4	53.1	48.3		49.8	77.4	60.6
max of 11 runs	66.0	21.6	91.8	34.9	71.3	72.2	39.6	51.0	75.3	45.8	96.3	62.1	57.1	46.0	57.3	50.9	58.1	52.2	80.5	63.3

Table 5.5: Result of bagging over 11 classifier trained on different distantly labeled sets. For comparison we show the minimum, average, and maximal results for these 11 runs.

can be successfully adopted for domains where named entity recognition and normalization is still an unsolved issue and the closed world assumption might be an unsupported stretch. This is important, as named entity recognition and normalization is a key requirement for distant supervision. Distant supervision is therefore an extremely valuable method and allows training classifiers for virtually all kinds of relationships for which a database exists. We have shown that results obtained without a manually annotated corpus are competitive with purely supervised methods. Thus the tedious task of annotating a training corpus can be avoided.

Five benchmark evaluation corpora – having diverse properties, annotated by different researchers adhering to differing annotation guidelines – provide a perfect opportunity to evaluate the robustness and usability of distant supervision. Our analysis reveals that domain knowledge such as interaction words or "negative" knowledge bases consistently improves results across all five corpora. Two instance filtering techniques (pos/neg-iword and neg-iword) perform comparably well on all five corpora and are therefore recommended for robust relationship extraction. Filtering of pos and negative instances (pos/neg-pairs) is not recommended, as it achieves only on one corpus superior results. Ensemble strategies such as bagging do not improve overall performance but have a positive impact on classifier robustness by decreasing the risk of selecting an under-performing single classifier.

Surprisingly, class imbalance seams to be a less pronounced problem in distant supervision as often observed for supervised settings. One possible explanation might be that due to the noisy data, a classifier is less prone to over-fitting.

5.3 Basic Graph Matching

In this section we analyze a different approach for PPI-extraction relying on distantly labeled data[6]. It differs in two ways from the previously discussed methodology using Support Vector Machines: First, it is a pattern matching approach and does not learn a statistical model. Second, we refrain from the closed-world assumption and derive patterns on positive instances only. In the following we explain the approach and compare

[6] Joint work with S. Pietschmann, H. Liu, and U. Leser

performance with the previously introduced approach.

5.3.1 Training Data Generation

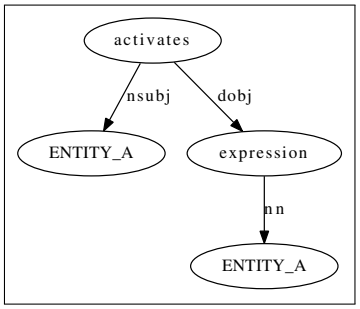
Distantly labeled data is generated following the methodology described in Subsection 5.2.1. In a nutshell: GNAT is used to recognize protein mentions in MEDLINE and PMC open access articles. Protein mentions are normalized to the Entrez Gene database. Protein pairs known to be interacting, according to the IntAct database, are than labeled as positive instances. All sentences containing at least one positive protein pair are converted into dependency trees using constituency parsing as an intermediate step.

5.3.2 Pattern Generation

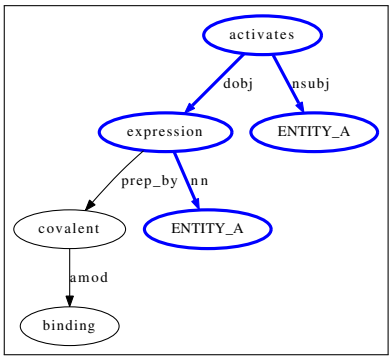
In a dependency tree, the shortest path between two tokens is often assumed to contain the most valuable information about their mutual relationship (Bunescu *et al.*, 2005). This is referred to as shortest path assumption and is frequently used in biomedical relationship extraction (see related work for PPI extraction in Subsection 2.5.1). Following this assumption we derive patterns from a dependency tree by extracting the shortest undirected path between the two named entities. In cases where more than one shortest path exists we use the union of all shortest paths. Hence, every shortest path is considered as a pattern. The initial set of patterns is denoted by S_{IP}. Entity mentions are blinded to ensure generalizability of learned patterns. Specifically, the mentions of the two proteins known to interact are replaced by the placeholder *ENTITY_A* and any additional proteins in the same sentence are replaced by *ENTITY_B*. In difference to the SVM approach we use only positively labeled instances (protein pairs) to derive patterns. The main reason for discarding instances derived by the closed world assumption is based on the the relatively high computational requirements for pattern matching. We will later see that negative instances build the majority of all samples and would therefore substantially increase matching time.

5.3.3 Basic Pattern Matching

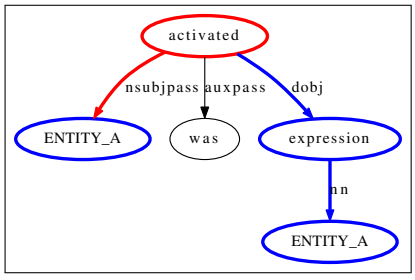
Sentences contained in the test corpus are converted into a dependency graph. We consider a pattern to match a subgraph of a dependency tree exactly, if all their nodes and edges match. This includes edge labels (dependency type), edge directions, and node labels (token and part-of-speech tag). Specifically we are looking for an injective mapping from our pattern to the dependency graph in question. An example is shown in Figure 5.5. The pattern shown in Figure 5.5(a) does not match the dependency graph shown in Figure 5.5(c) as one node (*activated*) and one dependency type (*nsubjpass*) differ. However, the pattern matches the protein pair in Figure 5.5(b). In order to increase F_1 we implemented some pattern processing rules as described in Pietschmann (2009). These rules will be subsequently described.



(a) Pattern derived from a distantly labeled sentence.



(b) Dependency tree for the sentence "Entity activates Entity expression by covalent binding.", where the pattern shown in 5.5(a) matches.



(c) Dependency tree for the sentence "Entity was activated by Entity expression.", where the pattern shown in 5.5(a) does not match.

Figure 5.5: The pattern depicted in Figure 5.5(a) matches the dependency tree in Figure 5.5(b) but not that in Figure 5.5(c). Matching edges and nodes are marked in blue, whereas mismatches located on the shortest path are highlighted in red.

Pattern generalization

It is a common practice in NLP to perform pre-processing on patterns to reduce corpus specificity. In particular, we perform stemming (G_{ST}) and replacement of interaction phrases by single words (G_{IW}). We summarize these two steps as *shallow generalization* steps. *Grammar-based generalization* encompasses the unification of dependency types (G_{UD}) and collapsing dependency links (G_{CD}). All four generalization strategies will be now explained in more detail:

1. **Word matching strategy (G_{ST})**
   Stemming is a commonly used technique to reduce the influence of inflectional forms of a token to its word stem. For example, the morphological variants "regulates, regulated, regulating, ..." share the word stem "regul". Replacing the word by the respective stem therefore decreases pattern specificity and potentially increases recall. In this work we used Porter stemming to derive word stems (Porter, 1980).

   As an alternative we also used the lemmatization library BioLemmatizer (Liu *et al.*, 2012). Lemmatizers perform a full morphological analysis to reduce tokens to their base form. For example, the word saw will be lemmatized to see or saw depending on whether the token was identified as noun or verb. In contrast, Porter stemmer will always return the token saw independently of the assigned POS tag.

2. **Collapsing interaction words (G_{IW})**
   Interactions between proteins can be expressed very diversely in natural language. Usually there is at least one word that semantically specifies the interaction. We refer to this word as *interaction word*. This is often a verb, such as "binds" or "phosphorylates", but can as well be a noun, such as "[induced] phosphorylation", or an adjective, such as "binding". The G_{IW} heuristic generalizes patterns by substituting all contained interaction words with generic placeholders. We assembled a list of 851 interaction words (including inflection variants) based on Temkin and Gilder (2003) and Hakenberg *et al.* (2006) that was further enriched manually. Based on POS-tags, interaction words are replaced by one of the three placeholders *IVERB*, *INOUN*, *IADJECTIVE*.

   We also experimented with a general interaction word placeholder ignoring the POS-tag of a respective word. In this case all interaction words are replaced with the same placeholder (*IWORD*). This strategy provides a higher level of generalization and handles incorrectly assigned POS tags.

3. **Unifying dependency types (G_{UD})**
   The Stanford typed dependency format[7] contains 55 different grammatical relations organized in a generalization hierarchy. Therefore, it is a natural idea to treat similar (*e.g.,* sibling) dependency types equally by replacing them with their common parent type. We manually evaluated all dependency types to assess whether

[7] Version 1.6

such replacements are viable. The final list of replacements is listed in Table 5.6. Note that we used the so-called collapsed representation of the Stanford dependency scheme. This means that prepositional and conjunctive dependencies are collapsed to form a single direct dependency between content words and the type of this dependency is suffixed with the removed word. For example, the dependencies `prep(located-2, in-3)` and `pobj(in-3, cytoplasm-4)` become collapsed to `prep_in(located-2, cytoplasm-4)`. In the G_{UD} generalizer, these dependency subtypes are substituted by their ancestors (*i.e.,* `prep`).

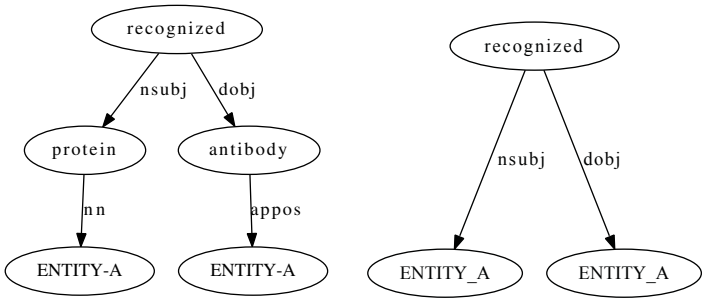
Dependency types	Common type
`subj`, `nsubj*`, `csubj*`	`subj`
`obj`, `dobj`, `iobj`, `pobj`	`obj`
`prep_*`, `prepc`, `agent`	`prep`
`nn`, `appos`	`nn`
`conj_*`	`conj`

Table 5.6: Unification of specific dependency types to a single common type by the generalizer G_{UD}. Note that dependency type `agent` is merged with `prep` as it is inferred for the preposition "by".

4. **Collapsing dependency links (G_{CD})**
   In addition to collapsing dependency types, we remove edges that most likely are irrelevant for describing PPIs. We focused on removing the dependency types `nn` (noun compound modifier) and `appos` (appositional modifier). These grammatical constructions have the same syntactic role but they carry somewhat different meaning. They function as noun phrase modifiers and often specify the subtype of an entity, which is irrelevant for our task. As these two dependency types convey no information about the interaction itself, the dependency and the corresponding noun can be removed; as long as the noun is not an entity. As an example, this generalizer is applied on the dependency parse tree of the sentence "*ENTITY_A* protein recognized antibody (*ENTITY_A*)" shown on Figure 5.6(a). The result of G_{CD} for this parse three is shown in Figure 5.6(b).

Pattern constraints

As previously discussed, our set of patterns also contains examples derived from sentences that do not describe an interaction. Such patterns lead to false positive predictions as they match dependency trees not mentioning an interaction. As a countermeasure, we define constraints a pattern has to comply with. Patterns not adhering to these constraints are removed from the pattern set, thus increasing precision. Filtering is performed before generalization, as generalization changes the pattern- and sentence-graph and may prevent identification and removal of spurious patterns. Standard heuristics for doing so are the exclusion of negation words (C_{NW}) and the restriction to patterns



(a) Original pattern (b) Generalized pattern

Figure 5.6: Dependency pattern before and after collapsing `nn` and `appos` dependency links using the generalizer G_{CD}.

containing interaction-related words from a predefined set (C_{IW}). On top of those previously known approaches, we developed two additional filters to leverage the semantic richness of dependency trees.

1. **Negation words (C_{NW})**
   Patterns containing negations potentially describe negative findings (*i.e.,* that two proteins do not interact with each other). Such pattern are removed to prevent wrong extractions. For negation words, we used the list of words described in Fundel *et al.* (2007). Additionally, patterns containing the dependency type *conj_no**, *conj_or*, or *prep_without* are removed as well.

2. **Interaction words (C_{IW})**
   Patterns without an interaction word might be to unspecific and potentially describe no interaction. Using the same list of interaction words as for the generalizer G_{IW} we remove all patterns without at least one occurrence of an interaction word.

3. **Dependency combination (C_{DC})**
   Interaction words are organized into the following POS categories: *verb*, *adjective* and *noun*. Based on linguistic considerations we define "dependency scaffolds" for the different POS categories. For example, we assume that interaction verbs describe an action that originates in one protein and affects the other protein. Obviously, the dependency combination `subj` with `obj` fulfills this consideration (for an example see Figure 5.6(b)). We manually evaluated a few dependency trees containing PPI for each interaction word category (verb, noun, adjective) and determined all combinations of dependency types that are valid for the given category. The resulting combinations are listed in Table 5.7.

4. **Syntax Filter (C_{SF})**
   A particular case in PPI extraction are sentences with enumerations, as shown in Figure 5.7. Such (possibly quite long; the longest enumeration we found contains

Part-of-speech	Dependency type combination	
Noun	`prep`	`prep`
	`prep`	`nn`
	`prep`	`amod`
	`nn`	`nn`
	`nn`	`amod`
Verb	`prep`	`subj`
	`prep`	`infmod`
	`prep`	`partmod`
	`obj`	`subj`
	`obj`	`infmod`
	`obj`	`partmod`
Adjective		`amod`

Table 5.7: Allowed dependency type combinations based on classes of POS classes (constraint C_{DC}). `subj` = {`nsubj`, `nsubjpass`, `xsubj`, `csubj`, `csubjpass`}, `obj` = {`dobj`, `pobj`, `iobj`} and `prep` = {`prep_*`, `agent`}

9 proteins[8]) enumerations greatly increase the number of protein pairs. Therefore, we developed a special treatment of enumerations based on dependency types. If two proteins have a common ancestor node connected by the same dependency type, we assume that those proteins do not interact with each other. Accordingly, we remove all such patterns. However, we also observed that sentences in which the common dependency type is `prep_between` or `nn` often do describe an association between the connected proteins. Accordingly, such patterns are retained.

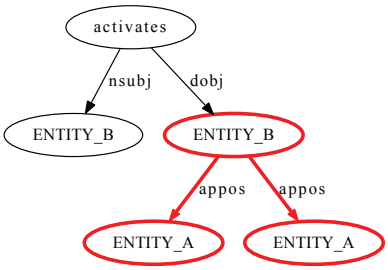


Figure 5.7: Dependency tree for the sentence "*ENTITY_B* activates *ENTITY_B*, *ENTITY_A*, *ENTITY_A*.". The investigated dependency pattern is highlighted in red. Application of C_{SF} removes this pattern.

[8]Sentence from PubMed-ID 19220217

5.3.4 Results

For evaluation we use the five manually annotated benchmark corpora: AIMed, BioInfer, HPRD50, IEPA, and LLL. All extracted patterns are matched against the dependency trees from these corpora. If at least one pattern matches, the respective protein pair is counted as *positive*. If no pattern matches, the pair is counted as *negative*. From this information we calculate precision, recall, and F_1.

372,083 patterns are collected from abstracts and 400,711 patterns are derived from PMC full-texts. In order to reduce runtime during the matching phase, we remove all duplicated (isomorphic) patterns. This procedure reduces the set of 772,794 initial patterns to 442,550 (57.5 %) unique patterns. Figure 5.8 shows the distribution of patterns by path length before and after removing isomorphic patterns. Unsurprisingly, longer patterns are usually more unique. For instance, 80 % of all patterns with length 2 can be removed, but only 12 % of all patterns with a length of 10.

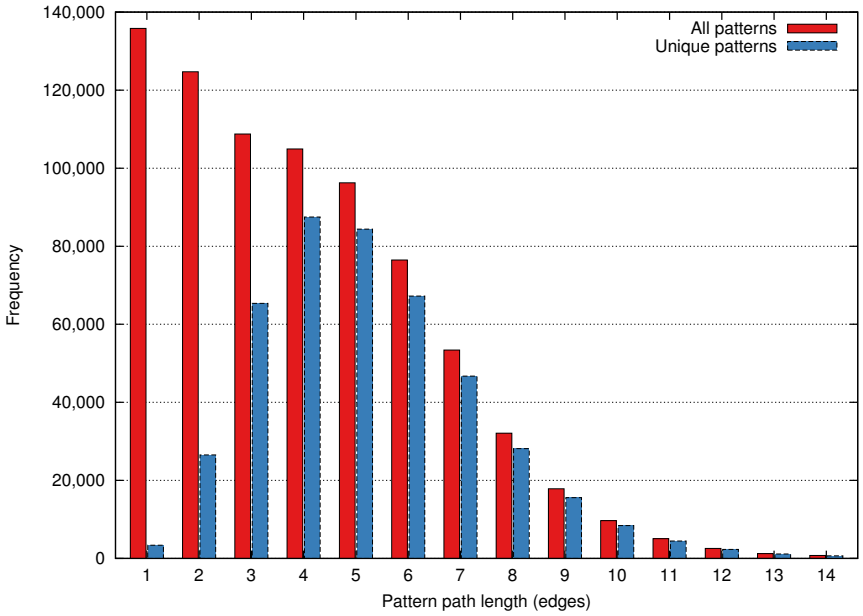


Figure 5.8: Distribution of all and unique patterns depending on pattern length (number of edges).

Pattern matching can be executed for each sentence separately, allowing parallel execution by multi-threading. Some sentences are, due to sentence boundary annotation errors, hundreds of tokens long. To decrease runtime, we restrict the runtime of the matching phase to 10 minutes for each individual sentence. After 10 minutes, we stop matching and check if any of the hitherto evaluated patterns matched the sentence.

Table 5.8 shows results using the initial pattern set as well as results for generalizations and constraints. We evaluate the impact of shallow and grammar based methods separately. $S_{shallow}$ encompasses stemming (G_{ST}), substitution of interaction words (G_{IW}),

interaction (C_{IW}), and negation word filtering (C_{NW}). While $S_{grammar\text{-}based}$ encompasses unification of dependency types (G_{UD}), collapsing dependency links (G_{CD}), the dependency combination constraint (C_{DC}), and the syntax filter (C_{SF}). In addition, results after application of all generalizers ($S_{generalizers}$), all constraints ($S_{constraints}$), and the combination of both (S_{all}) are also included. Corpus-specific results for the best setting in terms of F_1 (S_{all}) are shown later in Subsection 5.3.6.

Setting		AIMed			BioInfer			HPRD50			IEPA			LLL			
		P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	#
Baseline	S_{IP}	19.9	43.6	27.3	23.9	36.8	28.9	36.1	32.5	34.2	40.1	16.4	23.3	24.0	7.3	11.2	442,517
Generalizers	G_{ST}	20.6	46.9	28.6	24.3	38.5	29.8	36.7	33.7	35.1	42.6	18.8	26.1	29.1	9.8	14.6	—
	G_{IW}	20.7	51.5	29.5	26.3	44.6	33.1	36.8	39.3	38.0	53.7	34.3	41.9	45.3	23.8	31.2	—
	G_{UD}	20.3	47.7	28.5	24.3	39.4	30.1	35.2	34.4	34.8	42.9	18.8	26.1	30.9	10.4	15.5	—
	G_{CD}	20.0	48.6	28.3	25.3	43.9	32.1	37.3	36.8	37.0	51.0	29.9	37.7	31.2	15.2	20.5	—
Constraints	C_{NW}	20.3	43.1	27.6	24.7	36.3	29.4	36.8	32.5	34.5	41.0	16.4	23.5	25.0	7.3	11.3	396,496
	C_{IW}	48.2	29.3	36,4	51.8	10.3	17.2	85.4	25.2	38.9	75.8	14.0	23.7	88.9	4.9	9.2	323,756
	C_{DC}	37.0	27.7	31.7	47.3	17.3	25.3	84.0	25.8	39.4	90.4	14.0	24.3	85.7	7.3	13.5	210,599
	C_{SF}	22.2	42.3	29.1	32.4	32.6	32.5	38.8	31.9	35.0	43.7	16.4	23.9	27.9	7.3	11.6	250,929
Combinations	$S_{generalizers}$	23.4	**59.3**	33.5	34.9	**47.7**	**40.3**	39.5	**52.1**	45.0	55.4	**47.5**	51.1	55.0	**50.6**	52.7	—
	$S_{constraints}$	**61.2**	24.3	34.8	**85.9**	8.2	14.9	**97.6**	24.5	39.2	**93.8**	13.4	23.5	88.9	4.9	9.2	95,525
	$S_{shallow}$	40.6	34.4	37.2	59.7	16.6	25.9	69.4	30.7	42.6	78.4	31.3	44.8	85.4	21.3	34.1	—
	$S_{grammar\text{-}based}$	35.7	31.1	33.2	46.5	19.8	27.7	81.5	27.0	40.6	88.5	16.1	27.3	89.5	10.4	18.6	—
	S_{all}	38.3	37.1	**37.7**	61.6	25.6	36.2	80.8	36.2	**50.0**	85.7	39.4	**54.0**	**93.8**	37.2	**53.3**	—

Table 5.8: Performance of pattern sets for all five evaluation corpora. # denotes the unique pattern set size. Additionally to the different constraints and generalizers we evaluated the following settings. S_{IP}: initial pattern set without preprocessing, $S_{generalizers}$: all generalizers, $S_{constraints}$: all constraints, $S_{shallow}$: all shallow refinements (G_{ST}, G_{IW}, C_{NW}, C_{IW}), $S_{grammar\text{-}based}$: all grammar-based refinements (G_{UD}, G_{CD}, C_{DC}, C_{SF}), S_{all}: all refinements. Bold typeface indicate our best results for a particular corpus.

Generalizers

As can be seen in Table 5.8, all of the generalizers increase recall and even provide minor improvement in precision. For the combination of all generalizers ($S_{generalizers}$), an average increase of 24.1 pp in recall and 12.8 pp in precision was observed across all five corpora. Results of different generalizers are now discussed in more detail:

1. **Word matching strategy (G_{ST})**
   We first evaluated different token matching strategies. Beside exact token matching, we evaluated stemming and lemmatization. Initially, we also required the same POS tag between two tokens to match. We evaluated all combinations of token matching strategies (*i.e.,* exact, stemming, lemmatization) in conjunction with and without POS tag matching. Results for these six experiments are shown in Table 5.9.

The results indicate that utilization of POS tags in the matching phase leads to inferior results than strategies disregarding POS tags. Matching only the token without associated POS tag not only increases recall, but also leads to a higher precision in all experiments. We therefore ignore POS tags for matching in all following experiments.

Stemming as well as lemmatization show almost no effect on F_1 as long as POS tags are utilized during matching. Lemmatization without POS tags increases F_1 on average by 2.2 pp over the baseline using POS tags and exact token matching. On larger corpora lemmatization performs slightly better than stemming.

G_{ST} variant	POS	AIMed			BioInfer			HPRD50			IEPA			LLL		
		P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
Exact	✓	19.8	42.7	27.1	23.7	34.9	28.2	35.9	31.3	33.4	39.3	15.8	22.6	23.9	7.3	11.2
Exact	✗	19.9	43.6	27.3	23.9	36.8	28.9	36.1	32.5	34.2	40.1	16.4	23.3	24.0	7.3	11.2
Stemming	✓	19.8	42.9	27.1	23.7	34.9	28.2	35.9	31.3	33.4	38.8	16.1	22.8	26.0	7.9	12.1
Stemming	✗	**20.6**	46.9	**28.6**	24.3	38.5	29.8	36.7	33.7	35.1	**42.6**	**18.8**	**26.1**	**29.1**	**9.8**	**14.6**
Lemmatizing	✓	19.8	42.9	27.1	23.7	34.9	28.2	35.9	31.3	33.4	38.4	15.8	22.4	26.0	7.9	12.1
Lemmatizing	✗	**20.6**	**47.0**	**28.6**	**24.4**	**38.7**	**29.9**	**36.8**	**34.4**	**35.6**	42.2	18.5	25.7	27.8	9.1	13.8

Table 5.9: Performance of pattern sets for all five corpora using different token matching strategies (exact, stemming, and lemmatization). POS checkbox indicates if part-of-speech tags are also used during the token matching phase. Bold typeface indicate our best results for a particular corpus.

2. Collapsing interaction words (G_{IW})
From all generalizers, merging interaction phrases (G_{IW}) was proven to be the most effective, accounting for an average increase of 11.4 pp in recall and 7.8 pp in precision. As shown in Table 5.10, the variant, which merges all interaction phrases to a common word, is slightly superior to the variant in which interaction words are merged by POS tag.

G_{IW} variant	AIMed			BioInfer			HPRD50			IEPA			LLL		
	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
S_{IP}	19.9	43.6	27.3	23.9	36.8	28.9	36.1	32.5	34.2	40.1	16.4	23.3	24.0	7.3	11.2
Specific	20.6	51.0	29.4	26.2	44.2	32.9	**37.4**	**39.3**	**38.3**	53.2	32.2	40.1	43.4	22.0	29.1
General	**20.7**	**51.5**	**29.5**	**26.3**	**44.6**	**33.1**	36.8	**39.3**	38.0	**53.7**	**34.3**	**41.9**	**45.3**	**23.8**	**31.2**

Table 5.10: Results for collapsing interaction word variants (G_{IW}). Specific refers to the replacement of interaction words depending of the respective POS tag (*i.e., IVERB*, *INOUN*, and *IADJECTIVE*). General refers to the replacement of all interaction words by the generic placeholder *IWORD*. Bold typeface indicate our best results for a particular corpus.

3. Unifying dependency types (G_{UD}):

For the generalizer unifying dependency types (G_{UD}), each of the different variants was evaluated separately (see Table 5.11). The combination of all different variants leads, in comparison to S_{IP}, to an average increase of 2.8 pp in recall and 1.9 pp in precision. From the different variants, the unification of `prep` achieves the highest individual improvement in F_1 across all five corpora.

G_{UD} variant	AIMed			BioInfer			HPRD50			IEPA			LLL		
	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
S_{IP}	19.9	43.6	27.3	23.9	36.8	28.9	36.1	32.5	34.2	40.1	16.4	23.3	24.0	7.3	11.2
`subj`	20.2	44.6	27.8	24.0	37.0	29.1	**36.1**	32.5	34.2	40.1	16.4	23.3	25.5	7.9	12.1
`obj`	19.9	43.6	27.3	23.9	36.8	28.9	**36.1**	32.5	34.2	40.6	16.7	23.7	24.0	7.3	11.2
`prep`	20.2	45.7	28.1	**24.3**	38.2	29.7	35.9	34.4	35.1	42.8	18.5	25.8	29.6	9.8	14.7
`nn`	19.8	44.4	27.4	23.9	37.5	29.2	35.6	32.5	34.0	40.1	16.4	23.3	24.0	7.3	11.2
`sopn`	**20.3**	**47.7**	**28.5**	**24.3**	**39.4**	**30.1**	35.2	**34.4**	**34.8**	**42.9**	**18.8**	**26.1**	**30.9**	**10.4**	**15.5**

Table 5.11: Dependency type aggregations used in generalizer G_{UD}. `sopn` combines the dependency aggregations for `subj`, `obj`, `prep`, and `nn`. Bold typeface indicate our best results for a particular corpus.

4. **Collapsing dependency links (G_{CD})**:
   In the last experiment we evaluated the removal of specific dependency types from the dependency graph (see Table 5.12). Removal of compound noun modifiers (`nn`) provided a much stronger effect than the removal of appositional modifiers (`appos`). The best performance for G_{CD} can be observed when collapsing both dependency types.

G_{CD} variant	AIMed			BioInfer			HPRD50			IEPA			LLL		
	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
S_{IP}	19.9	43.6	27.3	23.9	36.8	28.9	36.1	32.5	34.2	40.1	16.4	23.3	24.0	7.3	11.2
`appos`	**20.0**	44.7	27.7	23.8	37.7	29.2	36.1	32.5	34.2	39.9	16.4	23.3	19.7	7.9	11.3
`nn`	19.8	47.4	27.9	**25.3**	43.0	31.8	**37.3**	**36.8**	**37.0**	**51.3**	**29.9**	**37.7**	**37.5**	14.6	**21.1**
`appos+nn`	**20.0**	**48.6**	**28.3**	**25.3**	**43.9**	**32.1**	**37.3**	**36.8**	**37.0**	51.0	**29.9**	**37.7**	31.2	**15.2**	20.5

Table 5.12: Impact of collapsing the dependency types `appos` and `nn` using generalizer G_{CD}. Bold typeface indicate our best results for a particular corpus.

Constraints

In contrast to generalizers, which alter patterns, constraints remove patterns from the pattern set. As shown in Table 5.8, application of all constraints ($S_{constraints}$) leads to an average increase in precision by 56.7 pp at the cost of a 12.3 pp decrease in recall. We discuss results of the different constraints in more detail:

1. **Negation words (C_{NW})**
   The shallow constraint C_{NW} eliminating patterns with negation clues has compa-

rably little impact and removes only a small fraction of all patterns (10 %). The removal of these patterns provides a rather small increase in precision (0.8 pp), accompanied by a small decrease in recall (0.3 pp).

2. **Interaction words (C_{IW})**
   The C_{IW} constraint removes all patterns without an interaction indicating word and is less conservative by removing more than 26.8 % of all patterns, trading off an increase of 41.2 pp in precision to a 10.6 pp decrease of in recall. In comparison to all other constraints, C_{IW} provides the strongest decrease in recall and the strongest increase in precision.

3. **Dependency combination (C_{DC})**
   With 52.4 % the dependency combination constraint C_{DC}, defining dependency scaffolds for different POS categories, removes the largest fraction of patterns. Although it removes substantially more patterns than C_{IW} the impact on precision, recall, and F_1 is less pronounced. This suggests that C_{DC} removes a large fraction of irrelevant patterns, but discriminative power is below that of C_{IW}.

4. **Syntax Filter (C_{SF})**
   The syntax filter constraint (C_{SF}) removes 43 % of the patterns and increases precision about 4.2 pp while recall drops moderately by 1.2 pp. In comparison to all other constraints C_{SF} provides the smallest decrease in recall across all corpora, indicating the high selectivity of this rule.

5.3.5 Error Analysis

We randomly picked 30 gold standard sentences (all corpora) containing false negatives and investigated all 72 false negative pairs included therein. For 29 of these pairs, possibly matching patterns were removed by C_{DC}, as the corresponding dependency combination was not covered in our rule set. Further 16 graphs passed the filtering, but our set of sentences contained no matching pattern. The third largest fraction of errors (13 cases) are pairs which, by our understanding, hardly describe an interaction. In 11 cases, the dependency parse trees are incorrect and therefore they do not provide the correct syntactic information. For 7 pairs, the shortest path covers insufficient syntactic information to decide whether two proteins interact. For instance, Figure 5.9 provides not enough information on the shortest path, whereas the second shortest path would provide sufficient information. Finally, three pairs were filtered by the C_{IW} filter, as their interaction words were missing from our list. We conclude that some constraints (especially C_{DC} and C_{IW}) are too aggressive. Relaxation of these syntactic rules should lead to higher recall.

We also analyzed the 30 patterns producing the most false positive matches. 20 of them contained an interaction verb, the remaining 10 an interaction noun. The 10 noun patterns produced more than twice as many false positives as the 20 verb patterns while matching about 50 % less true positives. The single noun pattern producing the most false positives (356) can be seen on Figure 5.10(a). Four other patterns can be seen as

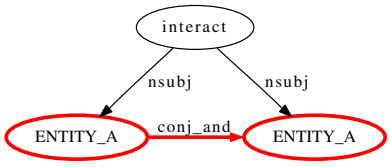
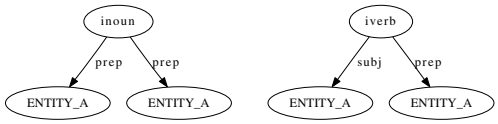


Figure 5.9: Example dependency parse where the information extracted by the shortest path (highlighted in bold red) is insufficient.



(a) Noun pattern (b) Verb pattern

Figure 5.10: Patterns producing the most false positives. Depicted dependency types are generalized according to G_{UD} and G_{IW}.

a variation of this pattern leading to a total amount of 732 false positives compared to only 172 true positives. This phenomenon is caused by the way in which generalizers and constraints were applied. The unification of different `prep_*` dependency types to the general `prep` (G_{UD}) makes some dependency type combinations indistinguishable, e.g. (`prep_to`, `prep_to`) and (`prep_to`, `prep_of`). The dependency type combination constraint (C_{DC}) would disallow a pattern containing the first combination, but as it is not applied in the matching phase, its benefits cannot be realized. A lesson learned from this example is that constraints should also be applied in the matching step: After a successful match, the constraints should be applied to the original un-generalized counterparts of the matching subgraphs. Similar conclusions can be drawn from examining the verb pattern producing the most false positives shown in Figure 5.10(b).

5.3.6 Comparison with other Methods

We compare the results of our best setting (S_{all}) with the cross-learning results for six[9] different kernels (Tikk *et al.*, 2010). We also show results of the rule-based system RelEx (Fundel *et al.*, 2007), as re-implemented by Pyysalo *et al.* (2008a). Additionally, we show results achieved by using distant supervision and a SVM classifier as introduced in Subsection 5.2. Detailed results are shown in Table 5.13.

The table indicates that on three out of five corpora our approach achieves the highest precision. For the remaining two corpora graph matching achieves an above average precision. Distant supervision using a SVM classifiers outperforms graph matching on four of five corpora.

[9]Due to the long training time authors provided only cross-learning results for six of nine kernels.

Method	AIMed			BioInfer			HPRD50			IEPA			LLL		
	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
Shallow linguistic (Giuliano *et al.*, 2006)	28.3	86.6	42.6	62.8	36.5	46.2	56.9	68.7	62.2	71.0	52.5	60.4	79.0	57.3	66.4
Spectrum tree (Kuboyama *et al.*, 2007)	20.3	48.4	28.6	38.9	**48.0**	43.0	44.7	77.3	56.6	41.6	9.6	15.5	48.2	**83.5**	61.2
k-band shortest path (Tikk *et al.*, 2010)	28.6	68.0	40.3	62.2	38.5	47.6	61.7	74.2	67.4	72.8	**68.7**	**70.7**	83.7	75.0	**79.1**
Cosine distance (Erkan *et al.*, 2007)	27.5	59.1	37.6	42.1	32.2	36.5	63.0	56.4	59.6	46.3	31.6	37.6	80.3	37.2	50.8
Edit distance (Erkan *et al.*, 2007)	26.8	59.7	37.0	53.0	22.7	31.7	58.1	55.2	56.6	58.1	45.1	50.8	68.1	48.2	56.4
All-paths graph (Airola *et al.*, 2008)	30.5	77.5	43.8	58.1	29.4	39.1	64.2	76.1	**69.7**	78.5	48.1	59.6	86.4	62.2	72.3
RelEx reimpl. (Pyysalo *et al.*, 2008a)	**40.0**	50.0	**44.0**	39.0	45.0	41.0	76.0	64.0	69.0	74.0	61.0	67.0	82.0	72.0	77.0
Distant supervision (SVM)	21.4	**91.3**	34.7	**70.9**	39.3	**50.6**	44.3	**95.1**	60.4	44.4	53.1	48.3	49.8	77.4	60.6
Distant supervision (pattern matching)	38.3	37.1	37.7	61.6	25.6	36.2	**80.8**	36.2	50.0	**85.7**	39.4	54.0	**93.8**	37.2	53.3

Table 5.13: Cross-learning results. Supervised classifiers are trained on the ensemble of four corpora and tested on the fifth one (except for the rule-based RelEx). Best results are typeset in bold.

5.4 Advanced Pattern Matching

The previous section evaluated the impact of different strategies for pattern matching. This included an analysis of different word matching strategies (*e.g.,* exact or stemming), different filtering steps to remove potentially faulty patterns, and different generalizers to increase recall. In this section we provide a deeper analysis of pattern properties for PPI extraction. We also describe a possible way to incorporate manually annotated text corpora to estimate individual pattern performance. Finally, we explain a versatile strategy to tweak precision/recall by using approximate subgraph matching without the need for manually annotated data. We first start with an evaluation of different pattern properties for PPI extraction.

Pattern length

First, we determine the impact of pattern length on prediction quality. Results, depicted in Figure 5.11, show that precision steadily increases with pattern length. Interestingly, the opposite effect has been described for patterns derived on the surface level for biomedical events (Nguyen *et al.*, 2010). Tikk *et al.* (2013) showed that the shortest path distance between two entities positively correlates with classification difficulty. In other words the longer the path distance the harder a protein-protein interaction will be to detect.

Number of entities per pattern

Second, we analyzed the performance of a pattern depending on the number of entities it contains. Obviously, every pattern contains at least two proteins (the two interaction partners), but with increasing numbers of proteins we observe a dramatic drop of precision and recall, as shown in Figure 5.12. Similar results have been reported by Nguyen *et al.* (2010) for shallow linguistic patterns.

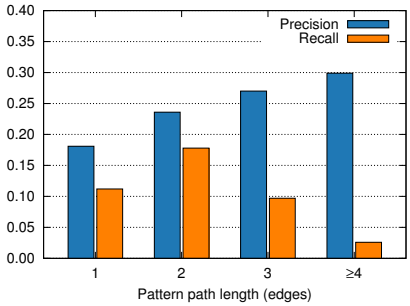


Figure 5.11: Pattern quality as a function of pattern length (number of edges).

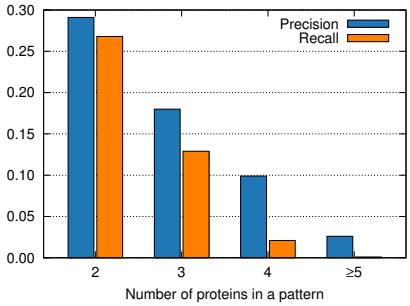


Figure 5.12: Pattern quality as a function of number of protein mentions in a pattern.

Increasing the amount of training data

One of the biggest advantages of distant supervision is that it avoids the need for manually annotated data. Distantly labeled data is comparably cheap and easy to produce provided a sufficiently large text resource and knowledge base. We therefore analyzed the impact of varying numbers of distantly labeled patterns, by sampling pattern subsets (see Figure 5.13). The relationship between pattern set size and performance has been modeled by linear regression using a logarithmic model. The model achieves a $R^2 = 0.975$ and the fit is highly significant ($p=1.628 \cdot 10^{-8}$) according to a F-test. The fitted model estimates that it would require 340,000 additional patterns to increase F_1 by 1 percentage point.

Shortest path assumption

The shortest path assumption is widely used in several relationship extraction algorithms (see Subsection 2.5.1). To test the reliability of the shortest path assumption we extracted the shortest path between all interacting protein pairs from the five training corpora and re-applied them on the same corpus (without applying any constraint or generalizer). The results of this experiment can be seen in Table 5.14. We observe very high recall values ($\geq 94\,\%$) for all five corpora. This shows that for most protein pairs a path between the two entities exist. This is not the case for entities sharing the same

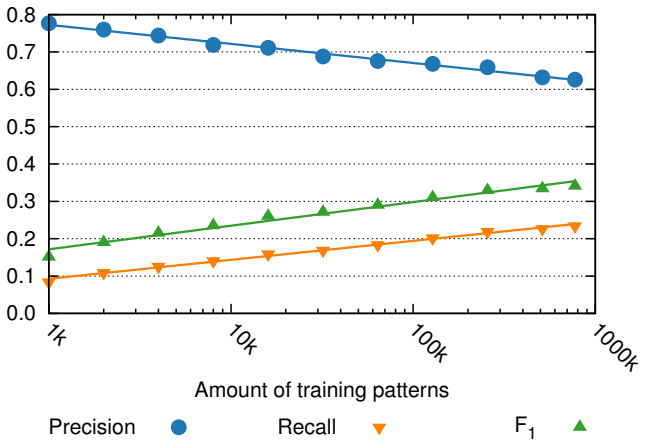


Figure 5.13: Evaluation of performance depending on the number of training pattern. Solid lines represent the fitted regression model for precision, recall, and F_1.

token (*e.g.,* BRCA1/2) and for rather complex/long sentences. More importantly, we observe surprisingly low precision values for the two largest corpora AIMed and BioInfer. It seems that the shortest path is not sufficiently discriminative and therefore leads to high false positive rates.

Corpus	Precision	Recall	F_1
AIMed	48.6	94.2	64.1
BioInfer	55.0	97.9	70.4
HPRD50	77.9	99.4	87.3
IEPA	89.6	100.0	94.5
LLL	98.8	100.0	99.4

Table 5.14: Application of patterns trained on the training corpus.

Distant supervision assumption

We previously argued that the distant supervision assumption is likely to be violated leading to noisy training data (see Subsection 5.1.1). In this experiment we evaluate if patterns derived by distant supervision achieve better results than arbitrarily selected patterns. We tested this by generating patterns for all found protein pairs without using a knowledge base. In other words we assume that all co-occurring proteins express an interaction between each other. This strategy extracts 29,319,149 patterns. To ensure comparability, patterns were sampled from abstract and full-text articles by keeping the same amount of patterns as used in the distant supervision experiments (*i.e.,* 372,083 patterns from abstracts and 400,711 from full-texts).

Before applying the patterns to the evaluation corpus we remove all duplicated patterns. Note that, the number of 520,697 distinct patterns is substantially larger (17 %), than the 442,550 distinct patterns generated by the distant supervision assumption. Due to computational constraints (one experiment takes up to 24 hours parallelized on 40 CPU cores), we evaluated the randomly selected patterns only for a subset of experimental settings. Results are shown in Table 5.15. Recall is usually several percentage points smaller using random patterns, despite the fact that the number of distinct patterns is approximately 17 % larger than patterns from distant supervision. On average we observe a 3.2 pp higher F_1 using patterns generated by distant supervision. The most notably contrast between the two set of patterns can be observed for S_{all} with a 6.1 pp difference in F_1. These experiments lead us to the conclusion that patterns derived by distant supervision are superior than randomly selected patterns. However, our generalizers and constraints considerably improve performance of randomly selected patterns.

Pattern ranking

Distant supervision does not depend on manually annotated data. However, given the existence of such data, we devised a simple strategy to incorporate manually annotated data using pattern ranking. Pattern ranking allows us to detect and remove error-prone patterns. We implemented a document-wise 10-fold cross-validation strategy, where patterns are first evaluated on 9/10th of all documents. For each rule we calculated precision, recall, and F_1 separately on the training set. Precision is set to zero for all patterns without any prediction on the training set. We than applied only patterns on the left-out documents with a precision equal or higher to a specific threshold. Results for varying thresholds are shown for AIMed and BioInfer in Figure 5.14.

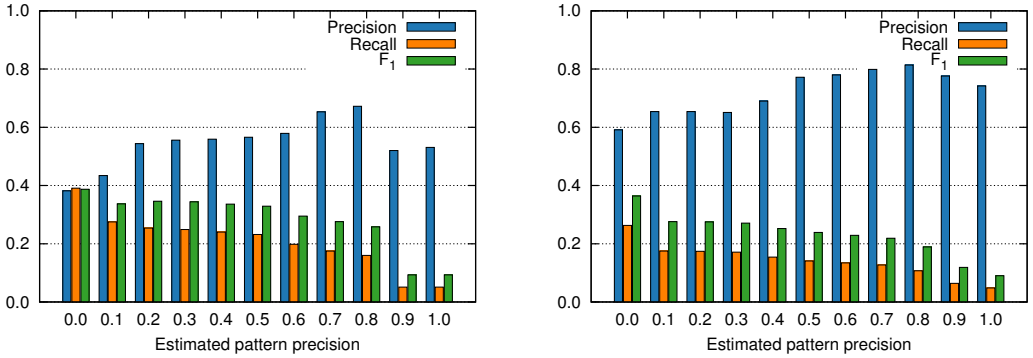
We observe a strong Pearson correlation ($r = 0.92$; $p=6.3 \cdot 10^{-5}$) between estimated pattern precision and the actually achieved precision. However, impact on recall is so strong that the highest F_1 is obtained by utilizing all available patterns (including patterns with an estimated precision of zero). Utilizing our currently best setting (S_{all}) we observe for all five corpora a lower recall than precision. In this setting recall has a much stronger impact on F_1 (see Section 2.3). However, by pattern ranking we can only increase precision, but not increase recall. For this reason pattern ranking is hardly effective to increase F_1. We will discuss an alternative measure to increase recall in the following subsection.

Approximate graph matching

Exact subgraph matching requires strict compliance between pattern and sentence subgraph. Liu *et al.* (2013a) introduced the concept of approximate subgraph matching (ASM) for dependency graphs which we use here and explain briefly. A pattern graph G_p is *approximately* isomorphic to a sentence graph G_s for a given threshold t, if the distance between two graphs is below that threshold. Here, subgraph distance is calculated for a specific injective mapping and represents the costs for transforming a rule graph

	AIMed						BioInfer						HPRD50						IEPA						LLL					
	Random			Distant			Random			Distant			Random			Distant			Random			Distant			Random			Distant		
Setting	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
S_{IP}	18.2	36.7	24.3	**19.9**	**43.6**	**27.3**	23.2	33.9	27.5	**23.9**	**36.8**	**28.9**	28.3	25.2	26.6	**36.1**	**32.5**	**34.2**	**44.0**	**20.9**	**28.3**	40.1	16.4	23.3	**28.3**	**9.1**	**13.8**	24.0	7.3	11.2
$S_{generalizers}$	21.8	51.9	30.7	**23.4**	**59.3**	**33.5**	31.6	41.6	35.9	**34.9**	**47.7**	**40.3**	36.6	41.7	39.0	**39.5**	**52.1**	**45.0**	55.3	41.8	47.6	**55.4**	**47.5**	**51.1**	**59.2**	35.4	44.3	55.0	**50.6**	**52.7**
$S_{constraints}$	**66.6**	20.1	30.9	61.2	**24.3**	**34.8**	**87.0**	6.9	12.7	85.9	**8.2**	**14.9**	93.5	17.8	29.9	**97.6**	**24.5**	**39.2**	93.0	**15.8**	**27.0**	**93.8**	13.4	23.5	**100**	**6.7**	**12.6**	88.9	4.9	9.2
S_{all}	**47.9**	30.3	37.1	38.3	**37.1**	**37.7**	**71.8**	16.3	26.6	61.6	**25.6**	**36.2**	**84.8**	34.4	48.9	80.8	**36.2**	**50.0**	**89.7**	33.7	49.0	85.7	**39.4**	**54.0**	**94.0**	28.7	43.9	93.8	**37.2**	**53.3**

Table 5.15: Results using 772,794 random patterns in comparison to using the same amount of patterns derived by distant supervision. Higher F_1 between these two pattern generation techniques are highlighted in boldface for each corpus and setting.



(a) AIMed (b) BioInfer

Figure 5.14: Evaluation of patterns with an estimated precision equal or higher to a specific threshold.

into the sentence subgraph. It is defined as the sum of three equally weighted distance functions:

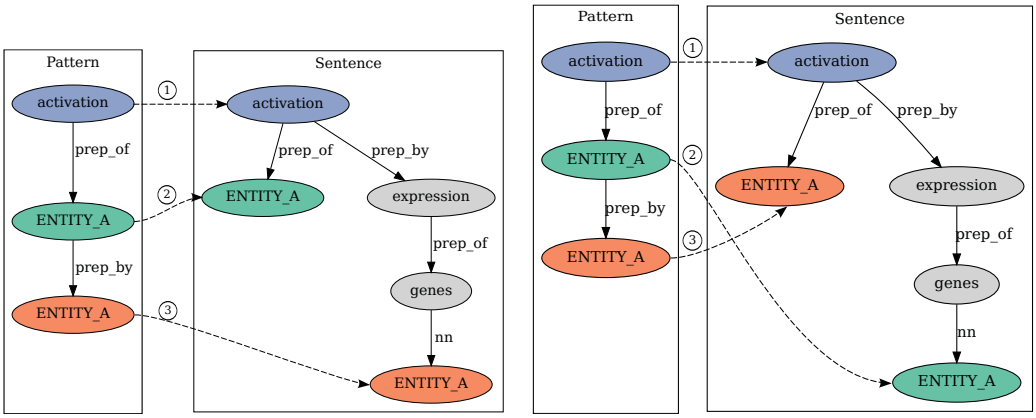
$$dist(G_p, G_s) = structDist(G_p, G_s) + labelDist(G_p, G_s) + directionalityDist(G_p, G_s) \tag{5.1}$$

structDist is defined as the difference of shortest path length for all pairwise matched nodes. The result is normalized by the average shortest path length of the sentence graph G_s between matched nodes.

labelDist calculates the number of different edge labels (*i.e.,* dependency types) between all pairwise matched nodes.

directionalityDist is defined as the number of different edge directions for all pairwise matched nodes.

The latter two distance functions are normalized by the number of edges contained in the pattern and sentence graph. Figure 5.15(a) depicts a possible injective mapping between a pattern and a sentence. The respective results of the three different distance functions are shown in Table 5.16. To find the minimal distance between two graphs the algorithm has to explore the entire search space (Tian *et al.*, 2007). Assuming that the pattern has n nodes and the sentence graph has m nodes where $n \leq m$ we end with $\frac{m!}{(m-n)!}$ injective mappings to evaluate. Due to the combinatorial explosion of possible mappings we use a heuristic to reduce the number of potential mappings. The algorithm considers only injective mappings where every node in the pattern graph exactly matches the respective node in the sentence graph. This heuristic substantially reduces the potential search space. For our example shown in Figure 5.15(a), the algorithm evaluates a second injective pattern where the target nodes of ② and ③ are exchanged (Figure 5.15(b)). Without this heuristic the algorithm would need to evaluate 60 different injective mappings. Liu *et al.* (2013a) published the implementation under a BSD license at `http://sourceforge.net/projects/asmalgorithm/`.



(a) First injective mapping (b) Second injective mapping

Figure 5.15: Two injective mappings between pattern and sentence graph.

Comparison	structDist	labelDist	directionalityDist
① – ②	|1-1|	0	0
① – ③	|2-3|	1	1
② – ③	|1-4|	3	3
$\sum$	4	4	4
Normalization	1+4+3	4+8	4+8

Figure 5.16: Subgraph distance for the injective mapping shown in Figure 5.15(a). The overall subgraph distance is $\frac{1}{2} + \frac{1}{3} + \frac{1}{3} = \frac{7}{6}$.

Results

For approximate subgraph matching we evaluate two set of patterns. First, we evaluate our best performing setting (S_{all}) using all constraints and generalizers in combination. Second, we evaluate $S_{constraints}$ where we only apply constraints and ignore generalizers. $S_{constraints}$ achieves a much higher precision than S_{all} using exact matching. ASM is expected to increase recall and it might be therefore beneficial to start with a set of few patterns of high quality. Because of the reduced number of patterns and the lower number of injective mappings to evaluate (due to the generalizers), we tested more ASM settings for $S_{constraints}$. Result of these two sets for varying numbers of ASM thresholds are shown in Table 5.16. For both settings we observe an increase in recall accompanied by a decrease in precision when incrementing ASM thresholds. Therefore, ASM allows us to emphasize our needs more towards precision or recall without the need of a manually annotated corpus. For several corpora, we observe at least one result outperforming the cross-learning results using the shallow linguistic kernel.

Setting	ASM	AIMed			BioInfer			HPRD50			IEPA			LLL		
		P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
cross-learning (Tikk *et al.,*)		28.3	86.6	42.6	62.8	36.5	46.2	56.9	68.7	62.2	71.0	52.5	60.4	79.0	57.3	66.4
Distant supervision (SVM)		21.4	**91.3**	34.7	70.9	39.3	**50.6**	44.3	**95.1**	60.4	44.4	53.1	48.3	49.8	77.4	60.6
S_{all}	0.0	38.3	37.1	37.7	61.6	25.6	36.2	80.8	36.2	50.0	85.7	39.4	54.0	93.8	37.2	53.3
	0.2	40.2	38.5	39.3	54.8	32.0	40.4	80.5	40.5	53.9	82.9	36.1	50.3	90.3	34.1	49.6
	0.4	31.0	56.1	40.0	46.0	42.3	44.1	59.1	69.9	**64.0**	60.5	72.8	**66.1**	71.5	81.1	**76.0**
	0.6	26.3	61.2	36.8	39.5	51.6	44.7	47.9	78.5	59.5	52.5	86.3	65.3	61.5	87.8	72.4
	0.8	23.1	66.1	34.3	35.0	59.2	43.9	41.7	86.5	56.3	47.0	**93.7**	62.6	57.5	**88.4**	69.7
$S_{constraints}$	0.0	**61.2**	24.3	34.8	**85.9**	8.2	14.9	**97.6**	24.5	39.2	**93.8**	13.4	23.5	88.9	4.9	9.2
	0.2	58.0	26.1	36.0	84.3	8.9	16.1	95.5	25.8	40.6	**93.8**	13.4	23.5	90.0	5.5	10.3
	0.4	49.7	37.9	43.0	72.6	15.5	25.5	87.0	36.8	51.7	87.4	22.7	36.0	**95.2**	12.2	21.6
	0.6	40.7	47.4	**43.8**	61.2	21.9	32.3	73.2	43.6	54.6	84.3	27.2	41.1	93.8	18.3	30.6
	0.8	30.3	63.6	41.1	43.6	38.6	40.9	61.6	60.1	60.9	72.1	44.8	55.2	88.5	42.1	57.0
	1.0	24.5	80.7	37.6	38.5	63.1	47.8	50.8	78.5	61.7	59.6	66.6	62.9	65.5	70.7	68.0
	1.2	21.6	88.2	34.7	34.6	**76.7**	47.7	46.2	89.6	61.0	53.6	77.9	63.5	52.5	84.8	64.8

Table 5.16: Results using approximate subgraph matching for two different sets of patterns. Bold typeface indicate the best results for a particular corpus.

5.5 Conclusion

This chapter discussed the applicability of distant supervision for protein-protein interaction extraction. We extensively explored two different approaches:

First, we applied a state-of-the art machine learning approach relying on Support Vector Machines to learn a statistical model. We explored two different heuristics to filter training instances and evaluated different aspects of classifier robustness. For instance, we showed that class imbalance has comparably small impact on F_1 within a class ratio between 0.1 to 10. With higher class imbalance the problem gets more pronounced. Similarly, we observe only little influence of increasing numbers of training instances. This indicates, that a comparably small size of 30,000 training instances is sufficient to

train a classifier on distantly labeled data. Finally, we showed that bagging increases robustness by decreasing the risk of selecting an under-performing single classier.

Second, we derived patterns from distantly labeled corpora. Patterns are generated for positively labeled instances only, therefore ignoring the closed world assumption. Constraints allow us to identify and remove faulty patterns and generalizers provide a tool to decrease pattern specificity. Approximate subgraph matching allows us to emphasize precision and recall. We further show that patterns from distantly labeled corpora provide better performance than randomly selected patterns.

Distant supervision using SVM achieves better out-of-the-box performance than our pattern based approach. However, using different processing steps we could substantially increase performance of pattern matching. An advantage of pattern matching is the interpretability of matched rules.

We showed that distant supervision achieves similar performance as classifiers trained on manually annotated training data evaluated in the more realistic cross-learning scenario. One of the biggest advantages of distant supervision is that it does not need manually annotated data. Considering the high efforts for manual curation, this is a particularly interesting feature. A classifier trained on distantly labeled data can be directly applied to label texts. This allows us to immediately assist curators in novel tasks. The result of such a curation effort is manually annotated data which could later be used to retrain the classifier using higher quality data.

5.6 Related Work

In this section we provide an overview of related work on distant supervision for relationship extraction. This is followed by an overview of methods utilizing graph matching for relationship extraction.

5.6.1 Distant Supervision

Craven and Kumlien (1999) originally proposed distant supervision in the context of biomedical relationships. The idea was adopted by Mintz *et al.* (2009) to Freebase relations (Bollacker *et al.*, 2008). Mintz *et al.* aggregate the feature space of positive instances expressing the same fact into one single datum. This aggregation step is usually refereed to as multiple instance learning (MIL). In difference to regular classification, MIL assigns one label to a bag of instances representing the same fact. Therefore, MIL provides one prediction for every given fact (bag of instances) instead of every instance separately. One disadvantage of this approach is that MIL points not to individual instances expressing the relationship. Also, Surdeanu *et al.* (2012) showed that classifying individual instances provides better results than MIL.

An important aspect of distant supervision is the comparably easy adaptation to different domains, which culminated in ideas to learn literally thousands of classifiers from relational databases such as Freebase (Mintz *et al.*, 2009; Yao *et al.*, 2010), Yago (Nguyen and Moschitti, 2011), or Wikipedia infoboxes (Hoffmann *et al.*, 2010). For instance, Hoffmann *et al.* (2010) use distant supervision to predict 5,025 different relationship types

from Wikipedia with an estimated F_1 of 61 %. Distant supervision has been successfully applied to several biomedical domains, including drug-drug-interactions (Thomas *et al.*, 2012b), protein-residue associations (Ravikumar *et al.*, 2011), or gene-drug, gene-disease and drug-disease relations (Buyko *et al.*, 2012).

Distant supervision relies on two noisy text mining techniques, *i.e.,* named entity recognition and relationship extraction. Errors of named entity recognition are propagated to the relationship extraction module, leading to worse predictions. To avoid this error propagation Yao *et al.* (2010) utilize factor graphs (McCallum *et al.*, 2009) for extracting named entities and corresponding relationship types in a joint fashion.

Bobic and Klinger (2013) incorporated active learning into the distant supervision scheme. They start with a subset of 200 manually labeled instances and train several classifiers using bootstrapping. The ensemble is used to select the most informative instances from a large set of distantly labeled instances. The authors observe a strong correlation between the labels assigned by the knowledge base and the confidence value assigned by the ensemble.

Intxaurrondo *et al.* (2013) analyze three heuristics to identify and remove noisy mentions generated by distant supervision. The first heuristics removes (positive or negative) facts which are too frequently found in the text resource. The rational is that at least one instance is wrongly associated. The second heuristic calculates pointwise mutual information (PMI) for every fact and removes facts with a too low PMI value. The last heuristic retains for every fact the 90 % most similar mentions. All heuristics together lead to an increase of 1.46 percentage points in F_1 on the dataset presented by Riedel *et al.* (2010).

5.6.2 Graph Pattern Matching

A early approach using pattern matching on dependency trees is RelEx (Fundel *et al.*, 2007). RelEx uses a small set of fixed rules to extract directed PPIs from dependency trees. Some of these rules also take advantage of dependency types, for instance, to properly treat enumerations. A reimplementation of RelEx was recently evaluated on the same corpora we used (see Table 5.13) and was found to be on par with other systems, though some of its measures were considerably worse than those reported in the original publication (Pyysalo *et al.*, 2008a). A notable difference to our approach is that RelEx rules were defined manually and are highly specific to protein-protein interactions. In contrast, we described a general method that performs pattern learning from automatically generated examples.

Liu *et al.* (2010a) utilize dependency tree patterns for event extraction on the BioNLP'09 corpus. Similar to our approach the authors also use shortest path assumption to generate patterns automatically. In difference to our approach, Liu *et al.* (2010a) learn patterns from manually annotated texts. The authors also experiment with different matching strategies, by aggregating different part-of-speech tags (*e.g.,* singular and plural nouns) or trigger words. This work is continued in the context of BioNLP'11 by Liu *et al.* (2011) where the authors remove patterns with low quality (*i.e.,* precision ≤ 0.25).

Liu *et al.* (2013b) use approximate subgraph matching to detect biomedical events for the BioNLP'13 challenge. Rules are subsequently ranked by precision and low ranking patterns are removed. In order to match tokens sharing the same meaning, the authors introduce a distributional similarity model (DSM). For instance, the words "interact" and "cooperate" can both be used to describe a protein-protein interaction. The authors implement the method from Pantel and Lin (2002) to find words sharing the meaning in a specific domain. Additionally, the authors derive patterns not only from the shortest path, but rather use all possible path as patterns. On the development set, the distributional model leads to an increase in recall, but drastically decreases precision. The all-path patterns increases F_1 moderately by 0.5 pp in comparison to the shortest path patterns. The authors removed DSM and all-path from prediction on the test set, as neither provides strong positive contributions.

MacKinlay *et al.* (2013) learn subgraph patterns from the BioNLP'13 training corpus. The authors follow a self-training inspired methodology to increase coverage (recall) of patterns. To this end, the authors incorporate the top-k results of TEES, a state-of-the-art tool for recognizing biomedical events, to infer additional patterns. For pattern matching the authors utilize the same matching strategy as Liu *et al.* (2013b) (*i.e.,* approximate subgraph matching and removal of low quality patterns).

Ravikumar *et al.* (2011) apply the distant supervision paradigm to identify protein-residue associations on MEDLINE. A notable feature is that the authors perform a "physical validation" to remove spurious protein-residue associations. This validation matches the residue and position against the protein in question. It has been shown that physical validation of protein-residue pairs achieves very high precision (Thomas *et al.*, 2011a); therefore leading to high quality positive training instances. Patterns are generated using the shortest path assumption for protein-residue pairs passing the physical validation step.

6 GeneView – End-user access to MEDLINE Scale Text Mining

Problems, such as synonyms, differences in word morphology, homonyms, and missing adherence of nomenclature aggravate recognition of named entities (see Subsection 2.4.1). This also impedes research: Ogino and Wilson (2004) pointed out that ambiguous nomenclature led to multiple discoveries of the same mutation by different groups, which could have been avoided by the usage of existing nomenclatures. Together with the rapid increase of biomedical literature (Hunter and Cohen, 2006), researchers face several problems when searching for relevant literature. For instance, Dogan *et al.* (2009) reported that over one third of all 58 million PubMed queries collected for March 2008 result in hundreds or even thousands of articles. Consequently, there is a growing body of research trying to provide improved retrieval for scientific texts to end-users (Lu, 2011). A pre-requisite for such advanced search features is high-quality named entity recognition and relationship extraction.

In previous chapters we covered different aspects of relationship extraction. These studies always evaluated results by using, relatively small, manually annotated corpora. In order to support biomedical researchers in satisfying their individual information needs, text-mining methods need to be applied to as many research articles as possible. Beside citations contained in MEDLINE, this encompasses also full-text articles, as they exhibit a much higher information content than abstracts (Schuemie *et al.*, 2004).

This chapter discusses the application of different information extraction components to all available citations in MEDLINE and full-texts in PubMed Central open access and is organized as follows: Section 6.1 describes the architecture and implementation of our large-scale semantic text mining engine GeneView. Section 6.2 covers computational resources needed to perform the individual text-mining steps. The user interface will be briefly described in Section 6.3. The shift towards large scale text-mining provides a setting to evaluate usability and performance of text mining tools on a larger scale without the need for relatively small gold standard corpora. Such an evaluation will be covered in Section 6.4, where we evaluate the reconstruction and expansion of an existing PPI network using text mining methods. This section also covers additional evaluations and applications using data provided by GeneView[1].

[1] Joint work with J. Starlinger, A. Vowinkel, S. Arzt, and U. Leser

6.1 Architecture

GeneView indexes all available citations from MEDLINE and full-text articles contained in the PubMed Central open access subset. Together with each article we store metadata such as authors, journal, MeSH terms, publication date, and others. All texts are imported into Lucene[2], serving as storage, query, and ranking engine. Upon import, texts are processed by a custom text-mining pipeline that incorporates a multitude of tools for pre- and post-processing and for the entity specific steps of named entity recognition, normalization, and relationship extraction. Information about all recognized named entities, especially type, position in the text, and normalized identifier are stored in a relational database to allow structured retrieval. The general architecture can be seen in Figure 6.1.

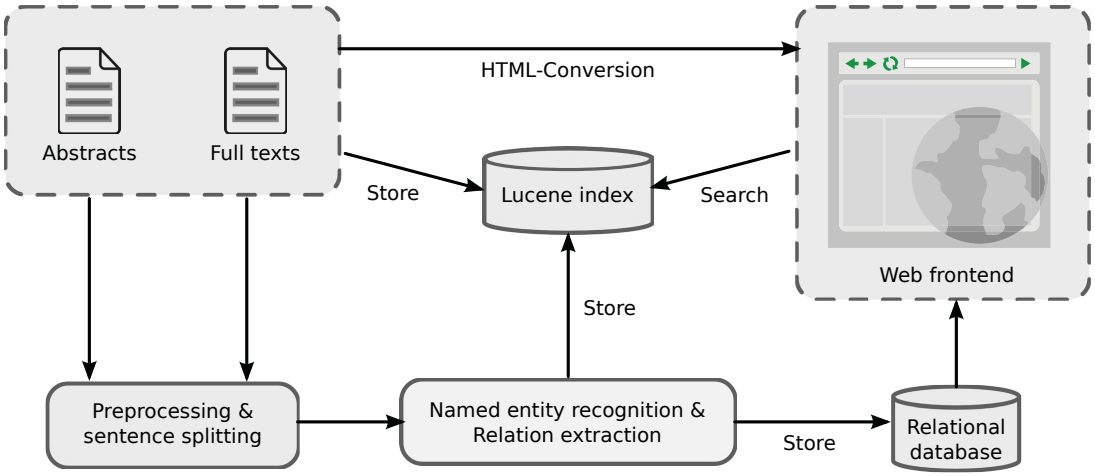


Figure 6.1: Architecture of GeneView.

6.1.1 Preprocessing

All citations are downloaded from the National Library of Medicine (NLM) as XML. Full text articles are converted into HTML for display in the GeneView web interface using XSLT scripts provided by the NLM[3]. This transformation generates HTML representations resembling the PubMed Central visualization and thus enables a similar user experience. Subsequently, the articles plain text is extracted: HTML specific characters such as "&amp" are replaced with the corresponding Unicode symbol. HTML elements (*e.g.,* </p> or <body>) are ignored and tables as well as references are removed. This extraction is necessary, as many text-mining components require cleansed text to work properly. Therefore, we need to store each full-text twice, once the HTML file for web

[2] `http://lucene.apache.org/core/`
[3] `ftp://ftp.ncbi.nih.gov/pub/archive_dtd/archiving`

representation, and once the raw text itself for internal processing. This duplication generates additional problems when it comes to exactly addressing character positions of text snippets for syntax highlighting. We therefore maintain a compressed mapping of character positions from the cleansed text back to the HTML file. For the visualization of abstracts, HTML is generated on the fly from the information stored in the Lucene index.

6.1.2 Information Extraction

Pre-processed articles are piped through a series of NER tools after sentence and abbreviation detection. For the individual steps we use specialized tools selected by a best-of-breed strategy. A problem with using independently developed tools is that they require different input format and date. Some tools require tokenized text, while others depend on unprocessed text because they perform their own tokenization. Similarly, some tools require text to be tagged with part-of-speech tags, while others perform POS tagging themselves. Relationship extraction depends on results from sentence boundary detection, gene name recognition, part-of-speech recognition, and, in our case, dependency parsing, which depends on the output of constituent tree parsing. Also, simple steps like abbreviation detection depend on preprocessing steps like sentence detection. The core of this workflow is depicted in Figure 6.2 and can be summarized as follows:

- First, section names are detected using an approximate dictionary covering the 200 most frequently mentioned section names. Sentence boundaries are detected using the free OpenNLP library[4] with models trained for biomedical articles (Buyko *et al.*, 2006). This model achieves an accuracy of 99 % on the Genia corpus.

- Abbreviation/long form mappings are identified using the algorithm by Schwartz and Hearst (2003), achieving 96 % precision and 82 % recall on a corpus of 1,000 MEDLINE citations.

- Gene name recognition and normalization is performed using GNAT (Hakenberg *et al.*, 2011). GNAT is based on custom dictionaries and conditional random fields (CRF) and normalizes gene mentions to Entrez Gene IDs. Using local context profiles and heuristics, GNAT tries to find the most probable Entrez Gene ID for a recognized gene. In uncertain cases GNAT associates gene mentions with more than one identifier. Because of different context profiles, a gene mention can be annotated in one sentence, but missed in another one. Annotations are therefore propagated to previously missed tokens including mentions of abbreviations and long-forms. The system was ranked among the first in several critical evaluations (Morgan *et al.*, 2008; Lu, 2011) and achieves, according to these assessments, a precision of 82 % and recall of 82 % for abstracts and precision/recall values of 54/47 % for full text articles.

[4] `http://opennlp.apache.org/`

- Single Nucleotide Polymorphisms (SNP) and other short sequence variations (*e.g.,* deletions, insertions, ...) are detected using SETH (Thomas *et al.*, 2014b). Mutation mentions following the latest recommendations are recognized by a Backus-Naur grammar proposed in Laros *et al.* (2011). To get hold of substitutions not following the nomenclature, SETH extends MutationFinder (Caporaso *et al.*, 2007a). Modifications encompass the detection of ambiguous amino acids (*e.g.,* Glx matches Gly and Glu), the detection of missense mutations (*e.g.,* Ala12Ter), and additional regular expressions for SNPs described on nucleotide level (*e.g.,* 650A>T). SETH achieves a precision and recall of 97.5 % and 80.7 %, respectively, on a test set of 508 abstracts. SETH also achieved competitive results in a recent evaluation comparing five SNP recognition tools in two different scenarios (Yepes and Verspoor, 2014). Subsequently, SNP mentions are normalized to dbSNP identifiers. This normalization procedure achieves a precision of 95.0 % and a recall of 58.0 % on a corpus of 296 documents (Thomas *et al.*, 2011d). Mentions of dbSNP identifiers follow a simple nomenclature (*e.g.,* rs334) and are recognized using regular expressions, achieving a precision of 98.2 % on a set of 100 randomly sampled documents.

- Species are identified and normalized to the NCBI taxonomy using LINNAEUS (Gerner *et al.*, 2010). LINNAEUS achieves a precision of 97 % and recall of 94 % on a test corpus of 100 full text documents.

- We recognize chemical compounds using ChemSpot (Rocktäschel *et al.*, 2012), a hybrid approach utilizing CRF for the detection of IUPAC-like chemical names and a custom dictionary for other chemicals, including trivial names, abbreviations, and molecular formulas. ChemSpot achieves a precision of 68 % and a recall of 69.5 % on the SCAI corpus (Kolářik *et al.*, 2008).

- Histone modifications are recognized using HistoNer (Thomas and Leser, 2013). HistoNer provides a set of 134 regular expressions and normalizes recognized entities to the Brno histone modification nomenclature (Turner, 2005). This approach achieves a precision of 94.4 % and a recall of 88.7 % on an evaluation corpus of 1,000 documents (Kolářik *et al.*, 2009).

- Drug names are identified using a custom CRF in combination with a drug name dictionary assembled from different drug related databases. The approach achieves a precision of 82.8 % and a recall of 74.2 % on the the AZDC corpus (Leaman *et al.*, 2009).

- Finally, mentions of "cell-types", "diseases", "enzymes", and "tissues" are recognized using dictionaries extracted from AliBaba (Plake *et al.*, 2006).

- Protein-protein and drug-drug interactions are extracted by training the APG and SL classifier on all available PPI/DDI corpora. Both methods have been explained in Section 2.5.1 and achieved outstanding results in different domains, including PPI extraction (Tikk *et al.*, 2010), DDI extraction (see Chapter 3), I2B2

challenge (Solt *et al.*, 2010), and the extraction of neuranatomical connectivity statements (French *et al.*, 2012).

A short summary of expected performance estimates is shown in Table 6.1. Considering the small size of available corpora, all mentioned evaluation values have to be considered as rough estimates.

Task	Tool	Precision	Recall	F_1
Abbreviation NER	ABBREV	96	82	88
Gene NER	GNAT	82	82	82
SNP NER	SETH	98	81	89
SNP normalization	SETH	95	58	72
Species NER	LINNAEUS	97	94	95
Chemical NER	ChemSpot	68	70	69
Histone NER	HistoNer	94	89	91
Drug NER	—	83	74	78

Table 6.1: Published performance estimates for named entity recognition tools integrated in GeneView.

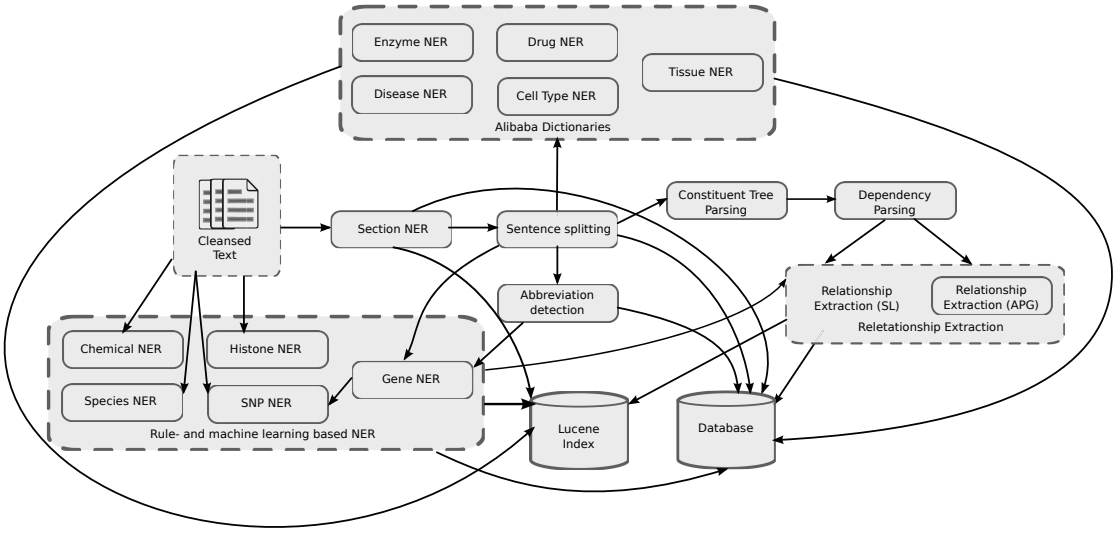


Figure 6.2: Pipeline of information extraction and NLP tools for creating the GeneView index.

6.1.3 Data Storage

All entities and relationships extracted by the text mining pipeline are stored in a relational database. Information stored for each entity mention includes the article ID, nor-

malized entity ID, annotated text span, and start/end-character position in the cleansed text. The article ID, which is the PubMed article identifier (PMID), links each mention to the corresponding document in the Lucene index. The normalized entity ID links a mention to additional information in external, type-specific data sources (*e.g.,* Entrez gene for genes, dbSNP for SNPs, ...). The annotated text span and the start and end positions precisely define the actual occurrence in the inspected document. This information is used for entity highlighting when visualizing single articles, which requires an additional step of mapping character positions as stored in the database to the HTML representation of the text created from the original XML files. For all relationship types (PPI and DDI), we store links to the two named entities, classifier confidence, and the respective sentence.

Document specific information is injected into the Lucene index for each entity type once the information extraction pipeline has finished. Thereby, Lucene can handle all ranking issues without a need to get back to the database; the database is only accessed for highlighting during web display (see above) and for assisting users in formulating queries. Here, GeneView provides on-the-fly auto-completion for entered tokens matching an entity name. This lookup issues a query to the database for each keystroke the user makes, which in turn requires a carefully indexed lookup table. We realize this lookup as a materialized view over the entity-specific annotation tables storing the original mention, its normalized representation and its corresponding identifier. Each entry contains the overall number of occurrences in the corpus, which allows to rank auto-complete suggestions by overall frequency.

6.1.4 Document Indexing and Ranking

Most ranking and filtering functions are implemented using Lucene. However, Lucene in the first place is not aware of entity frequencies within a document. Furthermore, a search for selected named entities is not a native feature of Lucene. To achieve this functionality, aggregated text-mining results have to be propagated into the Lucene index and represented properly to integrate them into customized ranking mechanism. For each article this encompasses the number of recognized distinct entities for each type as well as identifiers of recognized entities for each article section. The number of distinct entities of a specific type is used to filter articles without any entity of interest and to rank results by the number of distinct entities. Named entity normalization enables users to search for articles containing specifically this entity of interest (regardless of homonyms and synonyms).

For gene queries, the relevance ranking of Lucene is modified and a section specific ranking is applied (Jacob, 2010). Optimal section weights have been automatically determined using NCBI's gene2pubmed and are shown in Table 6.2. Gene2pubmed provides manually curated links between PubMed articles and the genes contained in them. Using this data, we set section specific weights such that a query for a curated gene in gene2pubmed ranks the corresponding articles in gene2pubmed highest. This strategy allows us to estimate and improve the mean average precision of gene queries. The automatically derived section weights meet our expectation in that, for instance,

sections such as *title* are highly ranked, while *materials and methods* receive low weights.

To allow users to focus on their particular set of genes, GeneView allows the definition of individual gene lists which later can be used to filter/rank articles of any query. In such cases, the query is expanded with the members of the gene set. Implementing this feature only requires functionality for storing and managing personalized gene lists, while their integration into the ranking can be achieved with standard Lucene features. Note that achieving this functionality manually would be hard, as such gene lists often contain dozens or even hundreds of genes (in case of genetically complex diseases such as cancer). It would be conceptually straight-forward to expand this feature to additional entity types, but therein one carefully has to balance functionality and simplicity of the user interface.

Another feature of GeneView important for users is "rank by entity count". To this end, we extract aggregated counts from the database and store them as additional metadata in a proper Lucene field attached to each document. At query time, one can tell Lucene to use the information in this field for ranking or filtering. This solution works equally well for all types of counts; however, for usability reasons we currently provide this functionality only for SNPs and genes.

Section	Boost
Title	3.0
Abstract	2.0
Introduction	1.0
Result	2.0

Table 6.2: Section weights for gene retrieval yielding the best performance on gene2pubmed. Sections not mentioned in this table received a weight of zero.

6.1.5 Visualization

For single article visualization, all contained entities and their spans are requested from the relational database. For each entity type a separate instance of the articles HTML representation is enriched with highlighting in a type-specific color. When displayed in the browser, these instances are overlaid to appear as a single document. The objective of this multi-layered approach is to allow collision free multi-entity annotation. For instance, a single entity may be (correctly) identified as a drug and a chemical, causing two overlapping annotations. As GeneView's highlighting are semi-transparent, the resulting overlap of layers will appear to the user in a different, mixed color, indicating the detected ambiguity. A drawback is the need to transfer each text to the user, *i.e.,* from server to client, multiple times within a single HTML document. While this strategy is unproblematic for abstracts, it does raise scalability issues for lengthy full texts in terms of the number of different entity types which can be included. For instance, GeneViews web page of a full text including five different entity types can reach a size of around 1MB.

6.1.6 Implementation

This subsection discusses some general engineering experiences when developing a MEDLINE scale search engine. One problem in the application of text mining tools to large collections is their instability in terms of achieved performance. NER and RE tools typically are evaluated on small gold standard corpora (GSC) only, which are also used to train the systems. Accordingly, the obtained measures are only valid for these GSC. We observed this problem especially for gene names, where community wide evaluations are often evaluated on document level and not on instance level. In GeneView this effect is counteracted by following the "one-sense-per-discourse" assumption (Gale *et al.*, 1992). This rule states that all mentions of a word in a given text tend to carry the same sense. We apply this rule as follows: First, recognized entities are propagated to the respective short or long form. Notably, this simple method adds 2.8 million gene mentions. Second, when a NER tool tags a given token (or set of tokens) and we observe this token again in the same text, we also tag it. The effect of this trick is even more pronounced, as it adds 24.4 million additional gene annotations. These two post-processing steps together are responsible for 47.5 % of all recognized gene mentions and have an enormous effect on the user-perceived recall and on subsequent relationship extraction. However, the propagation is not as simple as it appears, as one has to carefully decide when a subsequent match in a text is "good enough" for annotation. This is non-trivial, as, on the one hand, names for the same gene may differ slightly (*e.g.,* ABC-2 and ABC2 or TGD and TgD), while, on the other hand, slight variations in gene names may be decisive (*e.g.,* "Fas" and "FasL" are two different genes).

Another problem of large-scale text mining is that some errors are only observed on a small subset of articles, which makes detecting them very hard. Examples are:

- Our database implementation (MySQL 5.1.36) supports only Unicode encoding version 3.0 and therefore fails storing characters of higher Unicode versions; an error observed only twice in all articles.

- GNAT occasionally tags trailing spaces for some entities ($< 0.1\,\%$), leading to inconsistencies in visualization.

- The XML format of MEDLINE is continuously modified, leading to unexpected parser break-downs (which are spotted immediately) or scrambled HTML visualization (which we cannot detect automatically).

- For full texts, we keep the XML provided by the publishers to support a journal-specific visualization, leading to diversity in, for instance, the way formulas are represented: Some journals integrate formulas as figures, whereas others enforce the use of MathML, which is removed by our parser in the cleansing step.

- For constituency parsing, we apply the Charniak Lease parser (Lease and Charniak, 2005) using the McClowsky re-ranking model (McClosky *et al.*, 2006b) which is unable to parse 14,618 out of the total number of 8,131,441 sentences. The reasons for its problems are not clear yet. It is, however, noteworthy that the large

majority (14,546) of problematic sentences came from full-text articles, although the majority of sentences are from abstracts. Again, the original parser is trained on sentences derived from abstracts, which are known to be different from full-text sentences (Cohen *et al.*, 2010). This problem required changes in the source code of Charniak Lease parser, as the parser stopped after seeing a problematic sentence and did not continue parsing.

6.2 Computational Requirements and MEDLINE Scale Results

GeneView is regularly updated using a server with 40 cores at 2.4 GHz and 1 Terabyte main memory. Time intensive tasks, especially XML parsing, NER, syntactic parsing, and relationship extraction, are performed in parallel. The computational requirements it takes to rebuild GeneView on a single core are shown in Table 6.3. Overall, running the entire pipeline in this mode would require an estimated time of 145 CPU-days. The most time intensive task is syntactic and dependency parsing, although we actually only parse those sentences which mention at least two genes or two drugs. Of all our NER tools, gene NER is the most time intensive due to its sophisticated disambiguation strategy responsible for mapping a gene mention to its correct database identifier. Overall disc space requirement is about 77GB for the Lucene index and 63GB for the metadata and database.

	Processing step	Time [min]	Size [MB]
Preproc.	Text indexing	1,870	73,155
	HTML conversion	742	19,173
	Sentence detection	320	18,279
NER	Gene recognition	28,133	8,782
	SNP recognition	19,120	3,747
	Histone recognition	9,134	2,833
	Abbreviation detection	659	1,275
	AliBaba dictionaries	1,660	10,425
	Chemicals recognition	2,437	17,374
	Species recognition	1,582	8,932
	SNP normalization	512	3,747
	Parsing	125,347	38,649
RE	PPI extraction	13,120	29,483
DB	Database import	4,142	—

Table 6.3: Overview for time and space requirements to set up the GeneView repository. The required disk space for text mining results is based on database consumption. Overall CPU time required to build GeneView is 208,778 minutes equaling 145 CPU days.

Indexing text and entities

All previously described text mining tools are applied to a comprehensive repository of 22,957,879 citations, 586,547 with full-text body. More details about the underlying text resources[5] can be found in Table 6.4. This table shows that approximately 36 % of all available sentences come from PMC full text articles. After application of all NER tools, the GeneView repository contains more than 256 million entities for 10 different entity types. The distribution of all found entities is shown in Table 6.5.

Overall, 14,307,987 articles are annotated with at least one entity. The number of distinct entities per type ranges from 595 (cell type) to 134,587 (chemicals). For species, genes, and chemical compounds we find a high number of different named entities. We detect almost 1.6 million SNPs and are able to unambiguously associate 565,996 SNPs (35.5 %) with a dbSNP identifier, facilitating the search for SNP specific information. The most frequent entity type are chemical compounds, with about 41.6 % of all articles containing at least one chemical entity. Species, drug, and disease names also occur comparably frequent. For these four most frequent entity types and for PPIs, Figure 6.3 shows the distribution of mentions per publication year.

	Abstract	Fulltext	Total
Documents	22,957,879	586,547	22,957,879
Sentences	171,137,059	101,555,815	268,352,426
Average sentences per document	7.4	164.3	11.7
Tokens	2,642,894,842	2,277,486,807	4,842,103,615
Average tokens per sentence	15.5	23.4	18.0

Table 6.4: Number and proportions of articles in GeneView. Total represents numbers for all full-texts and all abstracts.

6.3 User Interface

GeneView provides a web-interface to make the extracted data searchable and accessible by end-users (see Figure 6.4(a)). GeneViews search bar, which is provided at the top of every page, allows users to issue keyword queries on all available text documents. This includes entity-specific search for recognized entities using standard identifiers, *e.g.,* Entrez gene ID for gene identification. The search bar offers an auto-completion function to make it easier to find specific identifiers. For instance, typing BRCA into the search bar yields suggestions for possible Entrez gene identifiers starting with BRCA (*e.g.,* BRCA1 and BRCA2). Additionally, the search form provides various options for result ranking and filtering. For instance, the user can choose to only include publications in the search result, that mention certain types of entities (*e.g.,* genes, SNPs, or chemicals). Figure 6.4(a) shows the result listing for a search for publications containing two specific

[5] As of 12/10/2013

Entity Type	Entities	Distinct entities	Articles
Cell-type	108,845	595	48,231
Chemical	87,974,672	134,587	9,565,995
Disease	49,734,744	31,613	9,800,063
Drugs	54,954,081	3,089	6,863,358
Enzyme	1,167,236	2,640	724,721
Genes	51,466,246	90,560	3,421,701
Histone-mod	127,925	946	11,771
SNP	1,594,709	68,160	242,350
Species	56,627,106	129,852	9,950,862
Tissue	12,678	133	10,698
Overall	256,493,289	445,315	14,307,987

Table 6.5: Overview of detected entities in GeneView.

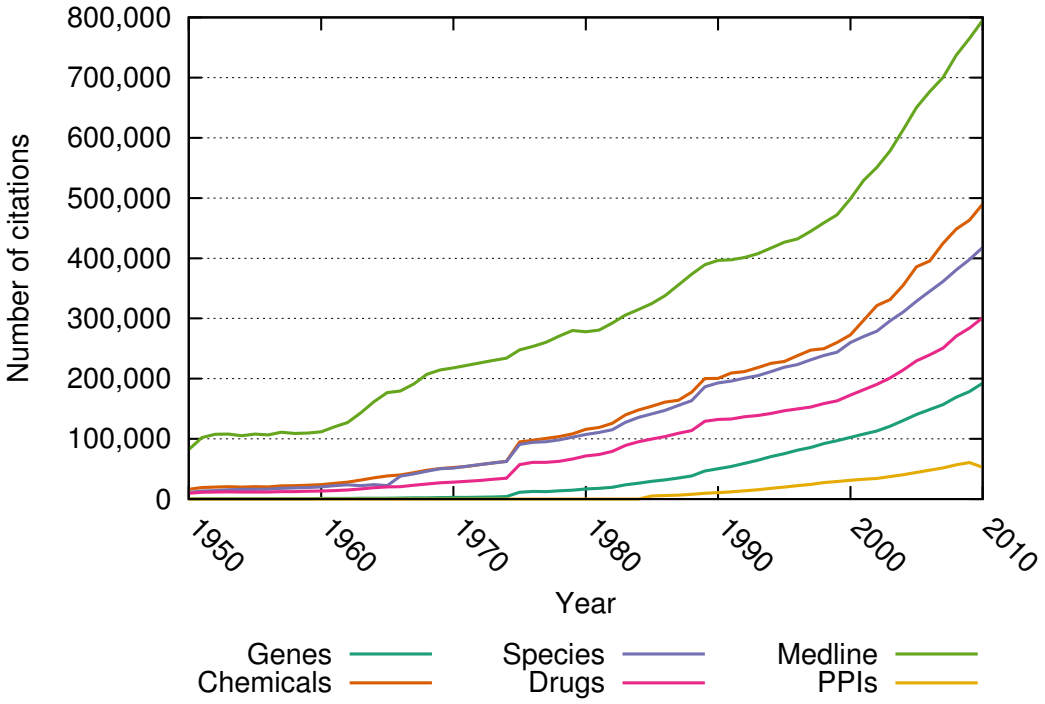


Figure 6.3: Number of citations tagged with at least one specific entity type.

genes represented by their Entrez gene ID. Here, the result is sorted by date of publication and has been filtered to show only articles that also contain at least one SNP.

Clicking on a specific search result shows the selected article together with annotations (see Figure 6.4(b)). Recognized entities are visualized by type-specific color highlighting. All entities are clickable to provide additional information such as link-outs to external databases. While GeneView extracts information about several types of entities to enable this type of multi-entity search, it does have special support for genes/proteins, where the on-click information contains links to several external reference databases of genes, pathways and protein-protein interactions. The pop-up also provides the option to search GeneView for articles describing PPIs in which the given gene/protein is found. GeneView also provides a summary of all entities found in the article (Figure 6.4(b), left-hand bar). This is particularly helpful when dealing with full text papers containing multiple mentions for various entities.

6.4 Applications

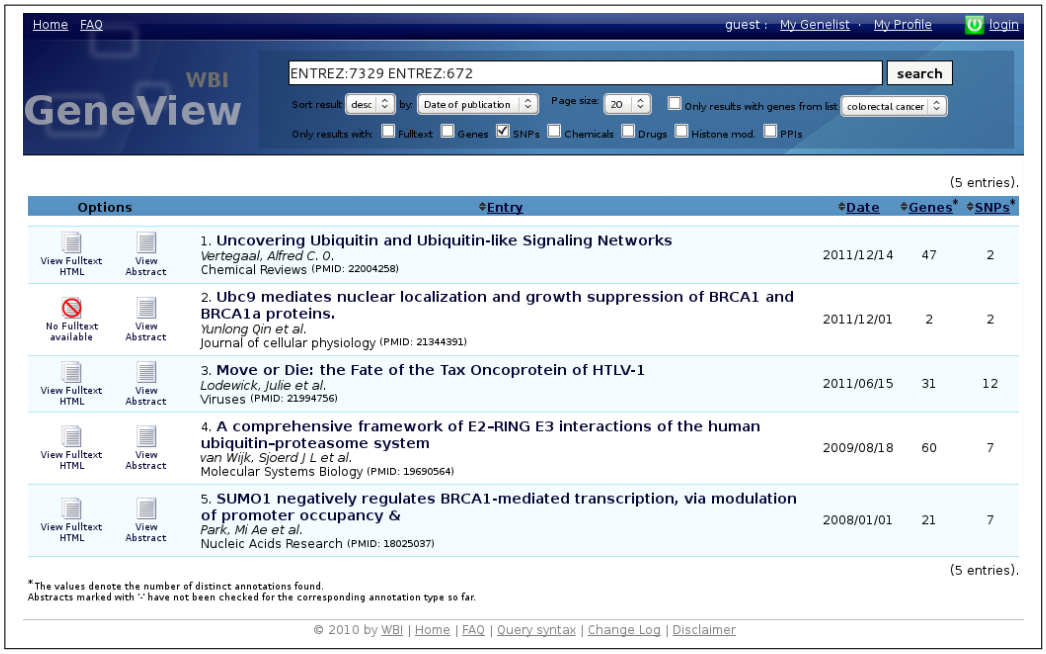
Most of the tools integrated into GeneView have been previously and independently evaluated on one or several data sets. However, evaluation corpora are usually small and reflect only parts of MEDLINE. Thus extrapolation of these results is difficult. Cohen *et al.* (2010) showed that linguistic aspects between full-texts and abstracts exhibit important differences. Most text-mining tools are developed and evaluated on abstracts and linguistic differences pose challenges in processing full-texts.

Therefore, we report on additional evaluations to show the utility of the data contained in GeneView. This ranges from general ideas like large scale evaluation and data analysis to more specific solutions like pathway reconstruction and data curation. Data contained in GeneView has also been used by Rodriguez-Esteban and Loging (2013) to quantify the increase of complexity of disease models over the past years using entropy models.

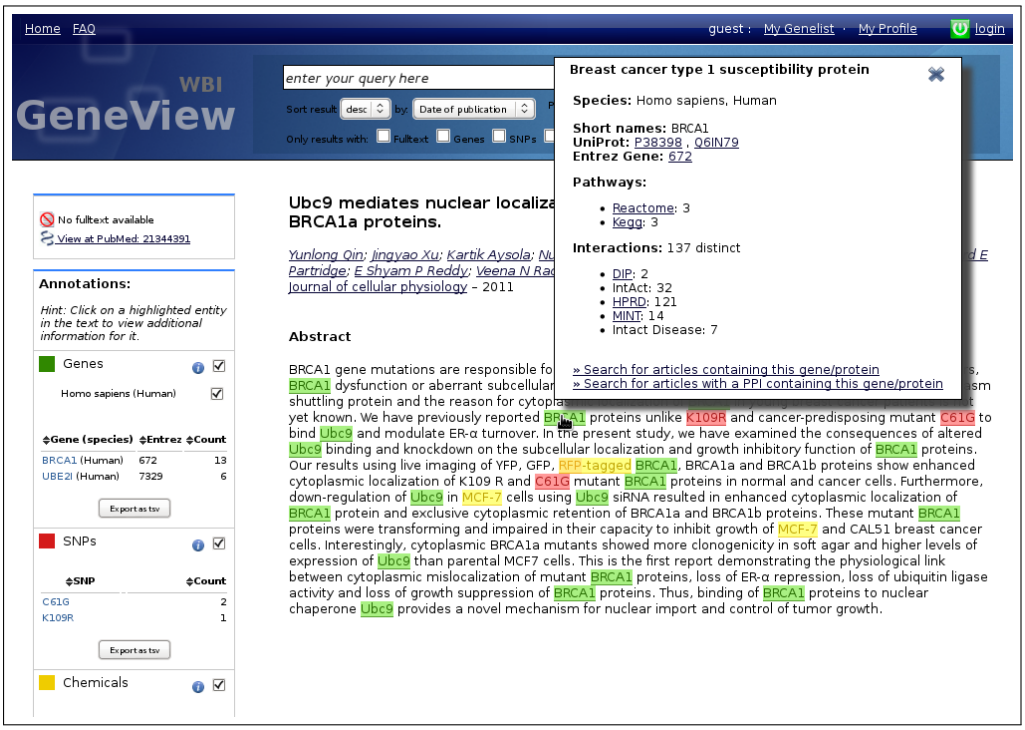
6.4.1 Extend of Annotation

In general we observe for all entity types a Zipfian distribution. For instance, Albumin is found in 70,994 different articles, but the majority (51.7 %) of all found genes are recognized in only five or less articles. In contrast, the 20 most frequent genes each appear in more than 21,000 scientific articles. Of those, 18 genes are human, indicating the focus of most gene-specific experiments reported in MEDLINE. This claim is supported by the observation that the MeSH term “human” is by far the most often annotated term in MEDLINE and occurs by one order of magnitude more frequent than the second species “rat”. For these 18 genes we performed a term enrichment analysis using David (Huang *et al.*, 2009). Terms like pathways in cancer, Jak-STAT signaling pathway, T-cell receptor signaling pathway, cytokine activity, growth factor activity, and negative regulation of cell death are significantly enriched (p-value < 0.01, after multiple testing correction). This suggests that immune response and cancer are major topics in biomedical research.

The database of GeneView can also be used to visualize trends in biomedical publications (Pfeiffer and Hoffmann, 2007; Palidwor and Andrade-Navarro, 2010). For in-



(a) Result of a search for texts mentioning two specific genes, filtered for articles containing SNPs, sorted by date of publication.



(b) GeneViews single article view of PubMed ID abstract 21344391. Inline entity highlighting is complemented by an overview of entities found in the text (left-hand bar). Highlighted entities provide pop-ups with additional information from external databases.

Figure 6.4: Screenshots of GeneView showing the search and result view.

stance, publications about the drug Thalidomide (Contergan) peak in early 1960s, when Thalidomide was identified as causing deformities in newborn babies (Daemmrich, 2002). In the late 1990s, Thalidomide completed a renaissance by becoming a FDA approved treatment for diseases such as leprosy and multiple myeloma (Rehman *et al.*, 2011). The distribution of articles mentioning the drug Thalidomide can be seen in Figure 6.5(a). Figure 6.5(b) visualizes the citation counts of the 8 most frequently mentioned genes over the last 40 years. It can be seen that genes such as TP53 or Albumin (ALB) received growing interest since their first discovery. Under the 10 most frequently mentioned genes we observe only one entity (TNFα) with a decreasing trend for the last years.

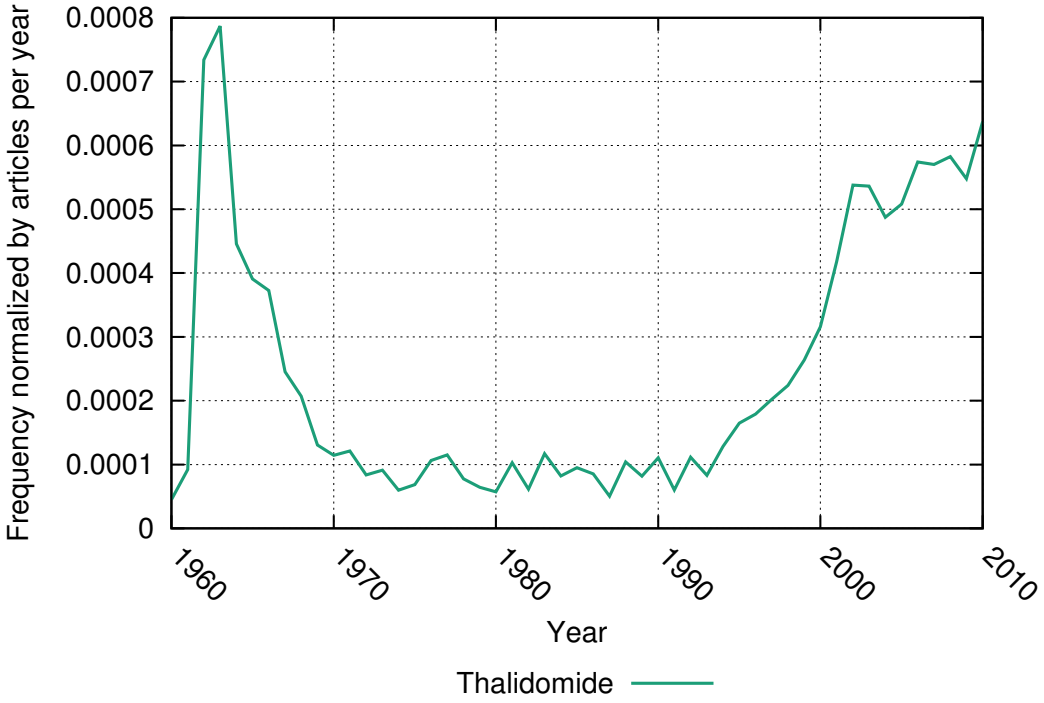
6.4.2 The Success of the Human Mutation Nomenclature

As another example, GeneView data can be used to analyze the acceptance of the human mutation nomenclature (Den Dunnen and Antonarakis, 2000) over the last years. The human mutation nomenclature has been introduced to reduce the ambiguity to describe mutations. For instance, Ogino and Wilson (2004) mention that the same SNP has been detected by several researchers due to inconsistent nomenclature usage. Similar problems have been reported by Berwouts *et al.* (2011), who analyzed laboratory reports for cystic fibrosis. The authors noted a gap between the nomenclature recommendations and complete implementation by genetic testing services as approximately 80 % of all reports used only outdated nomenclature to describe genetic variants. They also observed up to 20 different mutation names for the same mutation. Moreover, 5 % of variations used potentially malignant descriptors which might lead to misinterpretation of data. We analyze the advance of the human mutation nomenclature using our mutation recognition tool SETH. SETH distinguishes mutation mentions into three different categories:

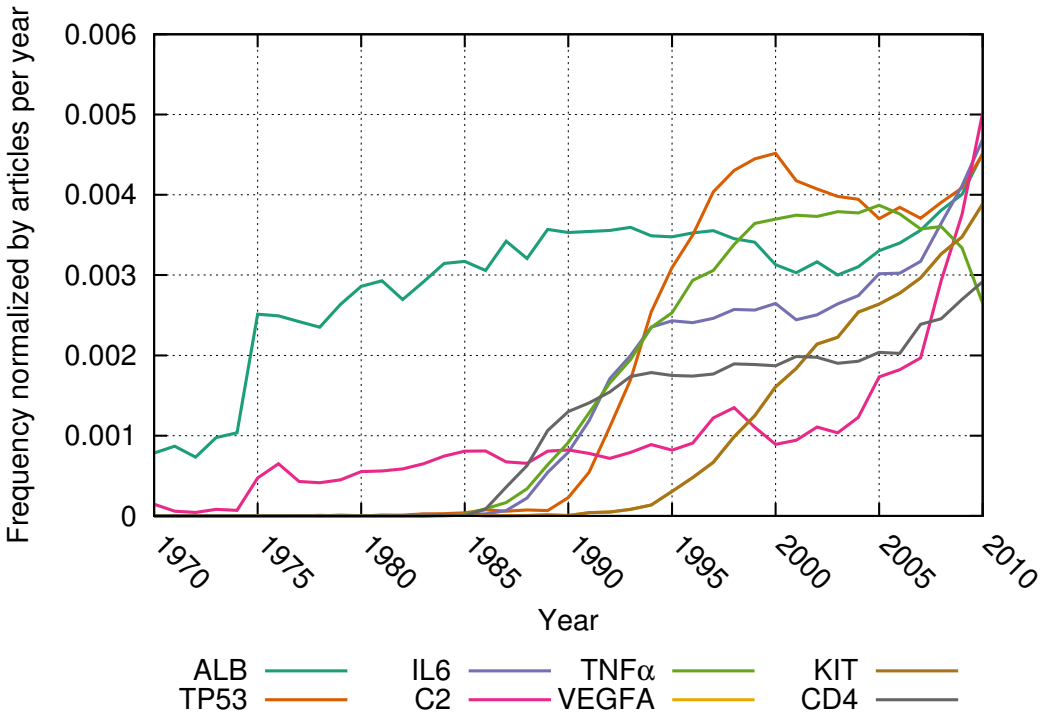
- HGVS: Mutations adhering the human mutation nomenclature (*e.g.,* p.Ile321Arg)
- Common: Mutations written in colloquial phrases (*e.g.,* Alanine substituted by tyrosine at position 12)
- dbSNP: Mutations described as dbSNP identifiers (*e.g.,* rs334)

The increase of publications mentioning mutations by any of the three forms is shown in Figure 6.6. It can be seen that most publications still use deprecated forms when referring to mutations. Interestingly, dbSNP identifiers seems to be more accepted than the use of HGVS nomenclature. A possible explanation is that the high expressiveness of the HGVS nomenclature is only worthwhile using when expressing complicated mutations. But approximately 90 % of all human mutations have been estimated to be single nucleotide polymorphisms (Collins *et al.*, 1998), which can be easily expressed without the human mutation nomenclature.

Finally, we investigated the distribution of mutation mentions for different journals since 2001 (when the nomenclature was reported first). For this analysis we ignored mutation mentions reported in the full-text to allow better comparability of different journals, as some journals do not participate in the full-text open access initiative of PubMed central. In total we found 201 journals reporting at least 500 mutations since



(a) Frequency of thalidomide mentions in MEDLINE for the last 50 years.



(b) Frequency of gene mentions in MEDLINE for the last 40 years.

Figure 6.5: Distribution of different entities over the last years. Frequency is divided by the overall number of articles per year.

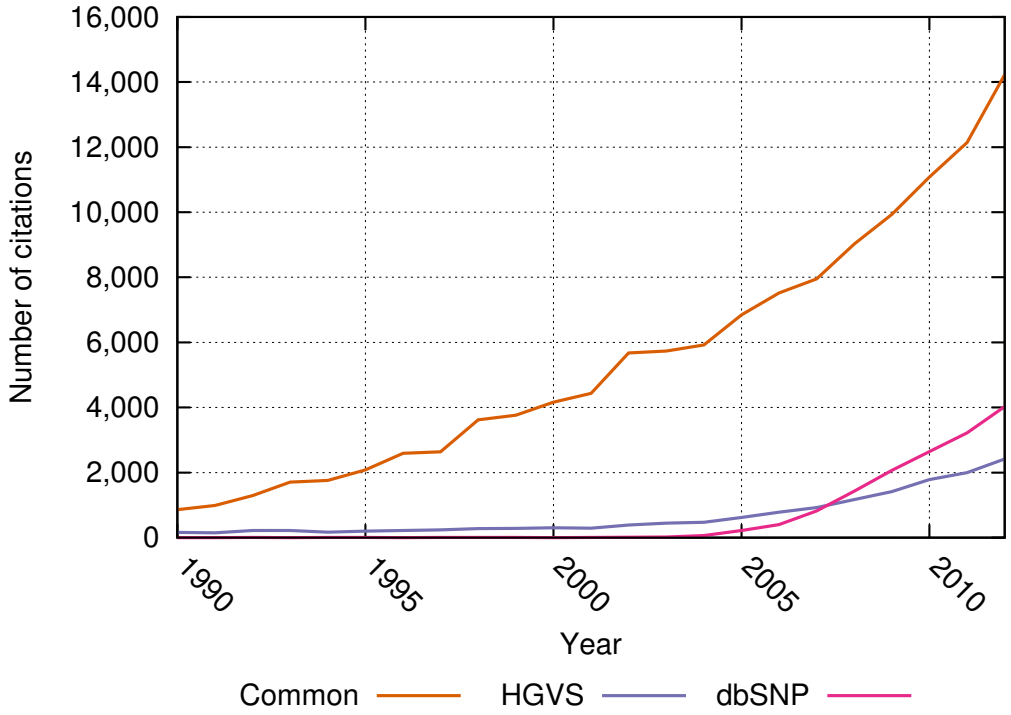


Figure 6.6: Increase of articles mentioning at least one of the three mutations since 1990. Common refers to mutations written in colloquial phrases, HGVS represents mutations adhering the nomenclature, and dbSNP specifies mutations described as dbSNP identifier.

2001. The 10 journals reporting most mutations are shown in Table 6.6. By far the most mutations have been reported in "The Journal of Biological Chemistry", however only a tiny proportion is actually described in the HGVS mutation nomenclature. The highest fraction of HGVS adhering mutation mentions can be found in "Human Mutation", as the journal requires scientists to utilize the latest nomenclature.

Journal	Mutations	HGVS	Percentage
The Journal of Biological Chemistry	25,686	116	0.5
Biochemistry	15,148	9	0.1
PLoS ONE	9,694	1,096	11.3
Human Mutation	7,597	5,305	69.8
Journal of Virology	5,626	6	0.1
Biochemical and Biophysical Research Communications	4,966	316	6.4
Proc. of the National Academy of Sciences of ...	4,438	92	2.1
Antimicrobial Agents and Chemotherapy	4,212	2	0.1
Journal of Molecular Biology	4,187	7	0.2
Biochimica et Biophysica Acta	4,131	216	5.2

Table 6.6: Distribution of mutation mentions for 10 journals with the most mentions since 2001. Column mutations represents the total amount of mutations written in coloquial form or adhering nomenclature. HGVS represents the amount of mutations written in HGVS mutation nomenclature.

6.4.3 Evaluation of Gene NER

Performance of named entity recognition is generally assessed using manually annotated corpora. Here we perform a large scale comparison between data provided by the NCBI and genes recognized by GNAT. NCBI provides curated links between genes and MEDLINE citations[6]. For both datasets (gene2pubmed and GeneView) we obtain a set of 2-tuples in form of {pmid, gene}. In other words a tuple indicates which gene occurs in which MEDLINE article. The overlap between the two sets is shown as Venn diagram in Figure 6.7. The significance of the overlap between the two set of tuples is derived using using χ^2-test (Pearson, 1900), where the null hypothesis is that the two sets are independent of each other. According to the χ^2 test the overlap between these two sets is highly significant (p-value $< 1 \cdot 10^{-16}$).

In a second experiment, we calculate for each gene the number of associated articles for data from gene2pubmed and from GeneView, respectively. Subsequently, we determine the correlation between these two results using Kendall's τ (Kendall, 1938) and calculate significance using Best and Gipps (1974) algorithm with the null hypothesis that no correlation exists. We observe a strong correlation ($\tau = 0.67$; p-value $< 1 \cdot 10^{-15}$) between GeneView and gene2pubmed, indicating a good overall agreement between gene name recognition and annotated data from NCBI.

[6] `ftp://ftp.ncbi.nlm.nih.gov/gene/DATA/gene2pubmed.gz`

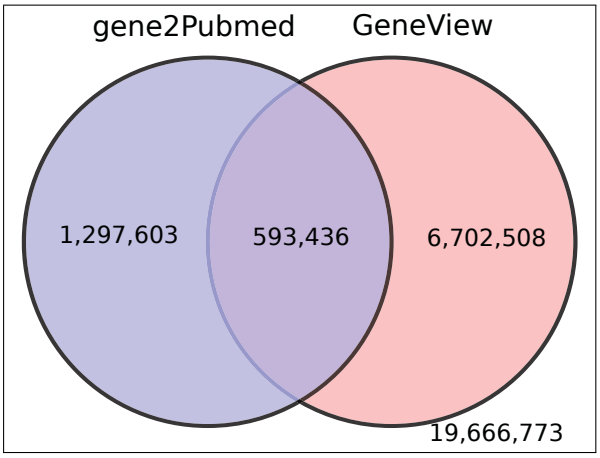


Figure 6.7: Overlap between genes recognized by GNAT and gene2pubmed for all of MEDLINE.

6.4.4 Pathway Reconstruction

We test how well PPIs extracted from GeneView coincide with known pathways[7]. To this end, we use all pathways contained in Kegg (Kanehisa *et al.*, 2012) and Reactome (D'Eustachio, 2011) as gold standard and compare the set of PPIs extracted for proteins within a pathway with those contained in the database[8]. Interactions found by our method but not contained in the specific pathway are considered as false positives. Interactions contained in the specific pathway but missed by text-mining are considered as false negatives. Pathways are obtained through the PPI- and pathway-database PiPa (Arzt *et al.*, 2011). We also compare the performance of the PPI extraction method with that of a simple co-occurrence based algorithm and investigate the impact of using full texts.

In 2,511,858 documents we observe 15,197,637 co-occurring protein mentions of which 3,921,267 (25.8 %) are classified as PPI by the SL algorithm. Using the approximately 3,9 million PPIs for the reconstruction of all 2,178 pathways from KEGG and Reactome we achieve a precision of 50 % and a recall of 5.1 % (averaged over all 2,178 pathways), equaling to a F_1 of 9.3 %. Recall generally is higher and precision lower for Kegg pathways than for Reactome pathways, indicating that Reactome pathways are more complete in terms of the literature-available data. Species-specific results are shown in Table 6.7, suggesting higher reconstruction quality for species such as human and rat. Note, that these five species (human, mouse, rat, fly, and arabidopsis) account for more than 85 % of all interactions in PiPa. Generally, these results seem better than that of previous similar studies. For instance, Rodriguez-Penagos *et al.* (2007) report a precision of 30–70% and a recall of 8–40% for reconstructing regulatory networks in bacteria.

We evaluate how the availability of full texts affects reconstruction performance. To

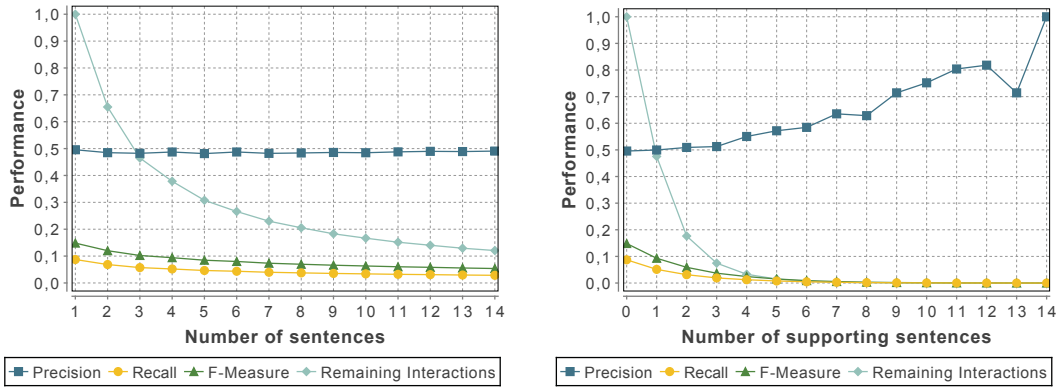
[7] Joint work with S. Arzt
[8] As of 08/31/2011.

Species	Precision	Recall	F_1	Pathways	Interactions
Human	45.8	14.5	22.0	527	27,132
Mouse	40.9	8.4	13.9	436	12,866
Rat	41.1	12.1	18.6	337	2,729
Fly	78.9	5.1	9.6	168	4,605
Arabidopsis	100.0	5.1	9.6	37	419

Table 6.7: Average reconstruction performance for different species ordered by the number of species-specific interactions. Interactions represents the amount of all interactions in the respective pathways.

this end, we repeat the previous experiment but consider only those articles for which a full text is present. We observe an increase of recall from 0.7 % to 2.5 % when moving from abstracts to full texts, clearly showing that full texts contain more information than abstracts. Precision slightly decreases from 59.9 % to 54.6 %, which is probably due to the fact that full texts also contain less relevant sections like material and methods. Filtering by section could probably counteract the decrease in precision.

We also compare our relation extraction algorithm with the common co-occurrence approach, which simply classifies all co-occurring protein mentions as interaction. Figure 6.8(a) illustrates how the number of co-occurrences supporting an interaction affects the reconstruction quality. Results for using only those protein pairs classified as interaction are shown in Figure 6.8(b). Clearly, the precision of pathway reconstruction does not correlate with the number of co-occurrences for one protein pair, whereas a clear correlation can be seen for positively classified pairs.



(a) Performance depending on the minimal number of co-occurring proteins.

(b) Performance depending on the minimal number of positively classified protein pairs.

Figure 6.8: Reconstruction performance by the minimal number of supporting sentences.

6.4.5 Extending the Circadian Clock

The previous evaluation considers pathway databases to be complete, as all interactions detected by GeneView that are not contained in a pathway are categorized as false positives. This assumption is a strong statement given the incompleteness of databases. For instance, Bauer-Mehren *et al.* (2009) showed that current pathway databases are highly incomplete when compared to manually curated pathways. Actually, one of the main reasons for using text mining in pathway reconstruction is the hope to find interactions not yet contained in a database.

We evaluate the usability of these "false" interactions to enrich an existing high-quality pathway for the mammalian circadian clock (Bozek *et al.*, 2009)[9]. The endogenous circadian clock regulates the timing of several biological processes allowing organisms to adapt physiology and behavior to daily rhythms. The circadian system is constituted by a genetic network of interconnected positive and negative feed-back loops which are able to generate oscillations in gene expression with a period of circa 24 hours (Zhang and Kay, 2010). Malfunctions of the circadian system are involved in many diseases (Takahashi *et al.*, 2008) and a detailed overview of the underlying genetic network is of major interest.

The currently known core of the circadian pathway consists of 121 interactions between 41 different proteins. Using GeneView we extract all PPIs between two circadian proteins and filter for those that are not yet contained in the pathway. To account for species specificity we map mammalian gene identifiers to Homologene clusters (Sayers *et al.*, 2012). Sentences containing potentially novel PPI are ranked by the confidence of the classifier (*i.e.,* distance to the hyperplane) and are subsequently evaluated by two domain experts[10].

GeneView contains evidence for 73 % of all 121 interactions described in the original circadian pathway. Additionally, the system suggests 190 novel interactions. Each novel interaction is supported by up to 851 sentences (in total 4,206 sentences). We reduce the number of sentences by ranking them by confidence and returning up to 5 sentences for each interaction. This reduces the amount of sentences to 580. Out of the 580 sentences, 209 (36 %) are classified as correct by the domain experts. Of the 371 misclassified sentences, 74 are considered as relevant but the sentence provides insufficient evidence for its finding. For 66 of these pairs the missing information could be found in the abstract and for 8 protein pairs it remained unclear even after reading the abstract. In total, we could enrich the pathway with 108 novel interactions supported by 132 MEDLINE references. The enriched circadian-core pathway is visualized in Figure 6.9.

6.4.6 Relationship Extraction using Co-occurrence

GeneViews repository currently contains nine different entity types, protein-protein interactions, and drug-drug interactions. So far each relationship type is extracted using a specific statistical model learned on annotated data. An alternative approach was

[9] Joint work with A. Relogio
[10] Angela Relogio and Alexander Frick

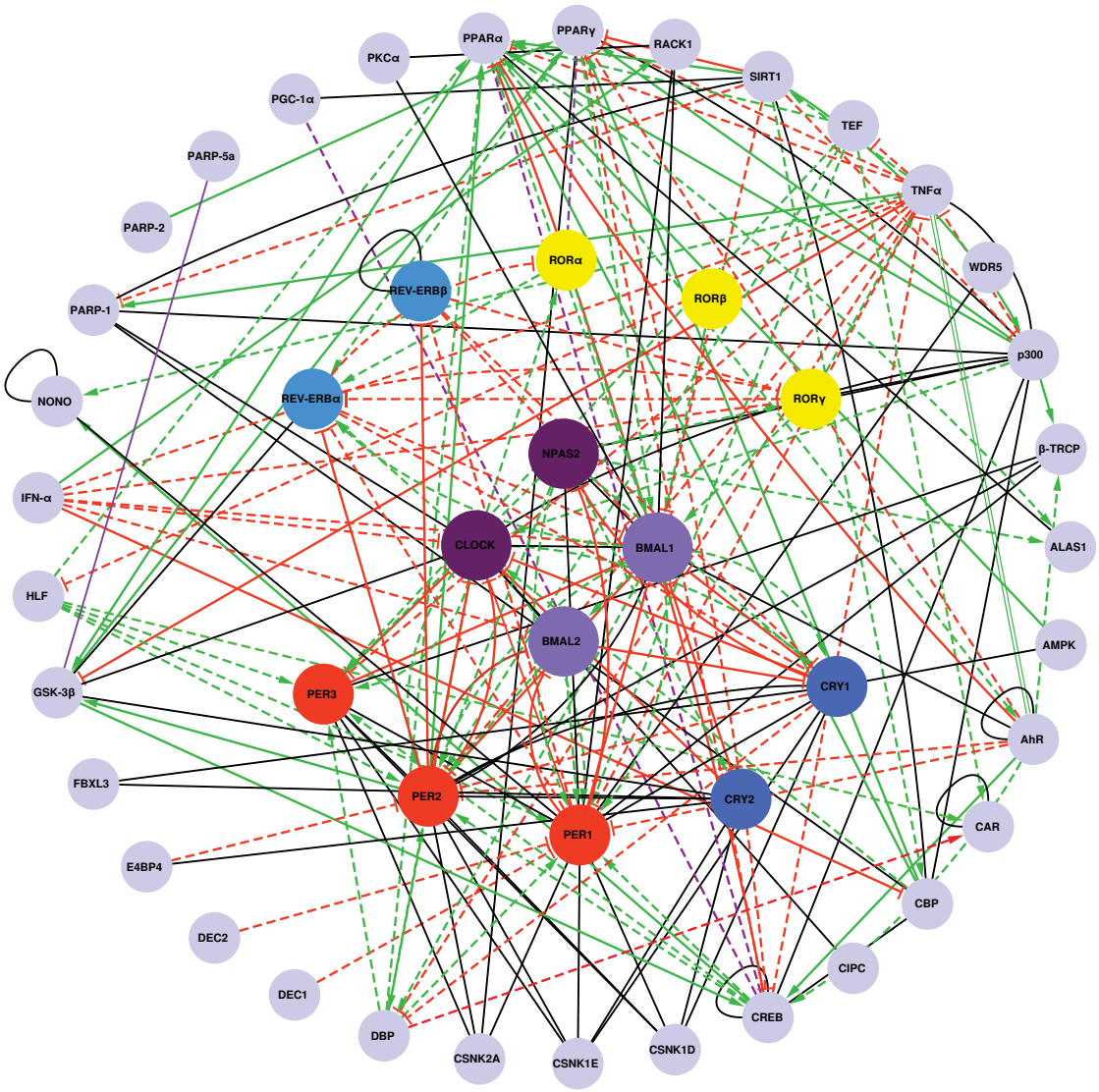


Figure 6.9: Comprehensive regulatory network for the mammalian circadian clock after annotation. In the center of the network we represent the main components of the basic feed-back loops. In the outer circle of the network we depict clock-regulated genes and proteins which feed-back to the core components and thereby influence the oscillations. Full lines, protein-protein interactions; dashed lines, protein-DNA interactions; red lines, inhibition interaction; green lines, activation interaction; black lines, other kinds of interactions.

presented in Section 5 where annotated data is build automatically using distant supervision. One disadvantage of distant supervision is the requirement of a database for every relationship type. This is less pronounced in other domains, where different relationship types are manifoldly contained in one knowledge base (*e.g.*, Wikipedia) (Hoffmann *et al.*, 2010). Here, we will show the usefulness of a simpler approach (co-occurrence) by the following example: Medical treatment of several cancer types depends on the mutational status of a patient. For instance, people suffering from colorectal cancer receive drug treatment depending on the mutations of several genes (Messersmith and Ahnen, 2008). Using GeneView, we searched for the 5 mutations most frequently mentioned with colorectal cancer. For these mutations we extracted the five most frequently co-occurring drugs. Results are shown in Figure 6.10. This plot shows several SNPs associated with treatment or progression of colorectal cancer. For instance, Val600Ala located on BRAF is a frequently used biomarker to predict reaction to Cetuximab and Sorafenib treatment. Other drugs associated to Val600Ala, such as Rasagiline and Dacarbazine are associated with the treatment of other malignant tumors.

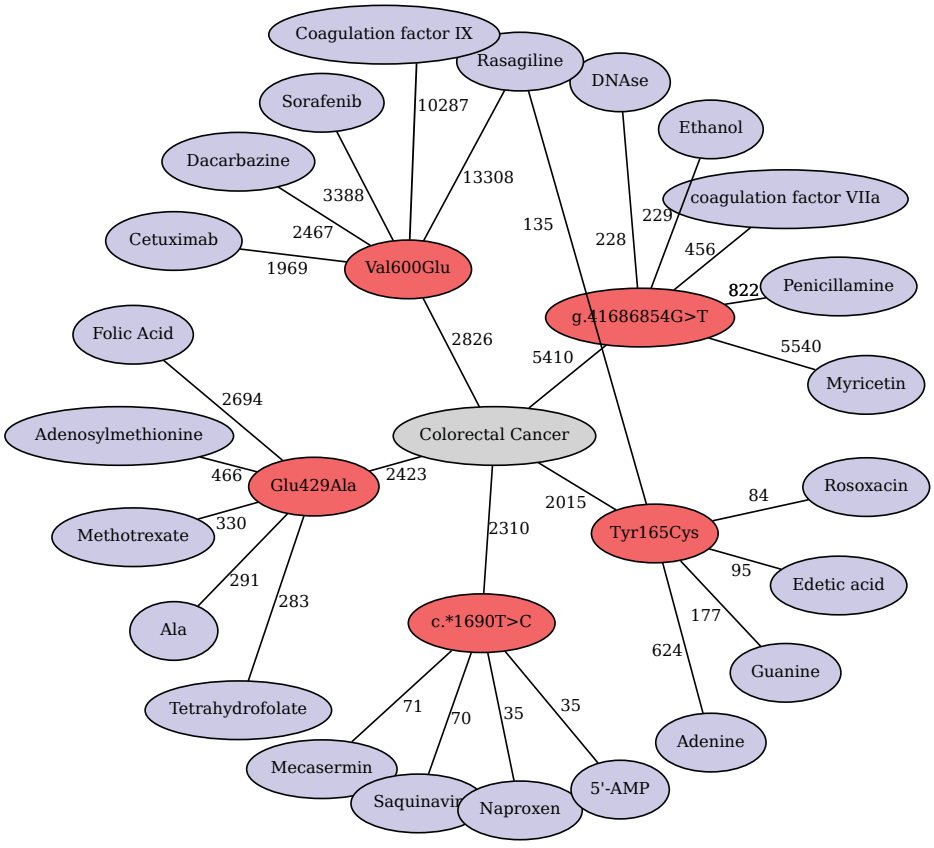


Figure 6.10: Co-occurrence graph for mutations associated with "colorectal cancer" and drugs associated with the respective mutations. Edge labels indicate frequency of sentence wise co-occurrence between the connected entities.

6.5 Conclusion

This chapter presented GeneView, an entity-centric search engine for the biomedical literature. The system encompasses several state-of-the-art NLP and information extraction tools whose output are stored in an information retrieval engine (Lucene) and in a relational database. We illustrated the usability of GeneView in various biomedical applications, including trend analysis, pathway reconstruction, and pathway augmentation.

Integration of heterogeneous specialized NLP tools lead to several problems, mostly due to changing requirements of data formats, multiple runtime dependencies, and execution environment. In particular, the lack of standards for representing annotated texts, which gives rise to many different ways to link annotations with text spans, creates the need to perform repeated format conversions and to keep multiple copies of the text, along with brute-force mapping tables. Several tools in this pipeline use a different format for the input text and the positional annotations it returns. In the future this problem might be alleviated due to recent efforts in defining standards to the community (Hellmann *et al.*, 2012; Comeau *et al.*, 2013).

6.6 Related Work

Several web-based tools have been developed for the extraction and presentation of semantic knowledge from MEDLINE. Most of these tools have a rather narrow and specific purpose, like retrieval of protein-protein interaction (PPI) data (Kim *et al.*, 2011b). We here only discuss those tools that are most similar to GeneView and refer to Lu (2011) and Rodriguez-Esteban (2009) for excellent reviews of this field.

iHop (Fernández *et al.*, 2007) enables the interactive navigation of MEDLINE sentences describing two protein mentions in conjunction with interaction specific key words. Entities different to proteins are not considered. AliBaba (Plake *et al.*, 2006) aggregates extracted knowledge across all results of a PubMed query and visualizes them as a graph. In difference to this work, GeneView focuses on individual documents. EbiMed (Rebholz-Schuhmann *et al.*, 2007) retrieves co-occurring entities for a specific query and ranks them by frequency. Like AliBaba and unlike GeneView, it provides aggregated results. Furthermore, GeneView uses a sophisticated machine learning technique to detect relationships instead of co-occurrences. Polysearch (Cheng *et al.*, 2008) and SciMiner (Hur *et al.*, 2009) offer a similar functionality as EbiMed, but use different extraction algorithms and significance tests. Facta+ (Tsuruoka *et al.*, 2011) enables the retrieval of indirect associations between biomedical concepts. PLAN2L (Krallinger *et al.*, 2009) can be used to rank sentences by relevance and visualize relations between entities focusing on *arabidopsis thaliana*. UKPMC (McEntyre *et al.*, 2011) extends the functionality of PubMed Central by using Whatizit (Rebholz-Schuhmann *et al.*, 2008) to recognize and highlight entities in abstracts. The system does neither highlight entities in full texts nor does it provide functionality to search with database identifiers instead of (possibly ambiguous) entity names. Finally, GoPubMed (Doms and Schroeder, 2005)

recognizes genes, gene ontology, and MeSH terms and presents search results using the structure behind these vocabularies. In contrast, GeneView recognizes a broader set of entity types but not gene ontology or MeSH terms, provides search facilities using unique database identifiers, and also finds relationships between proteins in texts.

More advanced information extraction techniques are also used by Björne *et al.* (2010), who performed extraction of nine biomedical events types on a sample of 1 % of all PubMed citations. This analysis has been later scaled to all of MEDLINE (Björne *et al.*, 2010) and later to PubMed Central full-texts (Landeghem *et al.*, 2013). While the system by Björne *et al.* extracts biomedical events, GeneView currently only annotates entities and binary relationships.

GeneView also differs from many of these tools in terms of user interface. Most systems present their results in form of ranked lists of entity pairs or single sentences. In contrast, GeneView presents its annotations in multiple ways. First, the entire article is shown with recognized entities being highlighted in different color codes. Next, all found entities and relationships are also presented as lists, a feature especially important for quick navigation in full texts. Finally, we provide annotations for all entity classes and relations as structured text files. Note that GeneView, in contrast to many other systems, includes the complete open PMC full text corpus on top of all MEDLINE abstracts. It also often annotates a broader set of concepts and uses more recent text mining methods. Annotations are provided as downloads to support the development of new applications by freeing developers of data analysis algorithms from the necessity to deal with a multitude of text mining packages.

7 Summary and Outlook

7.1 Summary

In this thesis we presented and evaluated different approaches for biomedical relationship extraction from texts. All proposed methods were evaluated on a set of manually annotated corpora. Methods have been assessed in various settings on corpora with particular properties, which allows us to investigate robustness in different scenarios.

Chapter 3 discusses our approach for drug-drug interaction extraction as originally proposed for the SemEval 2013 challenge. Our strategy implements a cascaded (coarse-to-fine grained) classification approach, which we evaluated on two different corpora (DrugBank and MEDLINE). The analysis reveals that training instances from DrugBank considerably help to improve DDI performance for MEDLINE articles. In contrast, the effect of MEDLINE articles for DrugBank is questionable and for some classifiers even misleading. Ensemble methods, combining the output of different classifiers, were used to improve performance over a set of eight individual classifiers. An important property of ensembles is that they improve robustness by reducing the risk of accidentally selecting an under-performing classifier. Stacked generalization outperforms majority voting by 1.1 pp on the evaluation corpus. More importantly, stacked generalization seems to be not affected by adding less informative classifiers, due to increased generalization capabilities over majority voting. In this intrinsic setting stacked generalization provides higher robustness than other methods.

Chapter 4 analyzes the impact of self-training to improve robustness for PPI extraction on texts with unknown characteristics. Robustness is an essential prerequisite for large-scale relationship extraction, as training corpora only partially reflect the target domain. 10-fold cross-validation suffers from the weakness that source and target texts potentially exhibit different characteristics, which is not properly reflected in cross-validation. Performance drops considerably when switching from an intrinsic evaluation to the more realistic extrinsic situation. We assess robustness of a classifier by performing cross-corpus experiments and improve extrinsic performance by self-training. The chapter investigates two self-training strategies, called self-only and self-enriched. In our experiments, both self-training strategies achieve higher robustness than a well performing baseline. In general, self-only achieves better results than self-enriched.

Chapter 5 analyzes the use of distant supervision for PPI extraction. Distant supervision automatically labels texts without manual intervention. In comparison to manual annotation, this strategy allows to increase training set size by some orders of magni-

tude. Corpora generated by distant supervision are inherently noisy, thus benefiting from robust relationship extraction approaches. In this chapter we compare two different approaches for protein-protein interaction extraction. The first approach learns a statistical model (SVM) on subsets of positive and negative instances. The second approach learns graphical dependency patterns from all positively labeled instances.

For the first model, we implement heuristics to remove likely mislabeled instances. We also analyze the impact of class-ratio in the distantly labeled training set as well as the amount of available training data. F_1 remains comparably robust with an average standard deviation of 2.6 pp for training class ratios between 0.1 to 10. We show that bagging, an ensemble learning technique, helps to improve classifier robustness by decreasing the risk of selecting an under-performing single classifier.

For the second approach, we define a set of pattern refinement strategies using generalizations and constraints. This strategy reduces noise and therefore improves robustness of learned patterns. We subsequently analyze different properties of patterns (*e.g.,* pattern length, amount of available patterns) on five evaluation corpora. Finally, we show that approximate graph matching allows us to emphasize our needs towards precision or recall.

Chapter 6 discusses the details for building the semantic search engine GeneView. It covers the architecture of GeneView and observed difficulties during implementation. A specific problem of large-scale text mining is that some errors are only observed on a small subset of articles, which makes detecting them very hard. We applied a cascade of state-of-the-art natural language processing tools on articles contained in MEDLINE and PMC open access. We sketched several use-cases utilizing data contained in GeneView. For instance, data contained in GeneView has been used to expand the circadian network (Relógio *et al.*, 2014). We also applied a similar workflow to extract regulatory relations between human transcription factors on all MEDLINE citations (Thomas *et al.*, 2014a). This procedure substantially decreases curation time by approximately one order of magnitude in comparison to a baseline working on co-occurrence.

7.2 Future Directions

In the following we discuss ideas for future directions concerned with relationship extraction.

7.2.1 Hybrid Approaches

A promising direction of supervised relationship extraction is the exploration of hybrid methods utilizing patterns in conjunction with machine learning. For instance, Bui *et al.* (2011) grouped protein-pairs according to their semantic properties and learned individual classifiers on the disjunct set of instances. Protein-pairs not adhering to any of the five previously defined groups are removed. This removal substantially alleviates the class-imbalance problem and improves classifier performance. Similarly, Chowdhury and Lavelli (2013b) introduce several heuristics to filter drug-drug interactions. For

instance, they remove all drug pairs referring to the same drug as self-interactions are extremely unlikely. Co-occurring drugs passing these heuristics are than used to train a classifier. Again, these heuristics lead to a more balanced class ratio between positive and negative instances.

A drawback of such rules is that they can only be applied when they are highly discriminative (*i.e.,* detecting either positive or negative instances with high precision). This is the case for the previously mentioned rules, but most heuristics are less discriminative and provide only vague clues. Instead of filtering training instances prior to classification, we propose the incorporation of these clues into the feature space of a classifier. This strategy allows the classifier to learn the importance of these clues (potentially in combination with other clues). For instance, we could introduce a feature indicating if the shortest path contains an interaction noun. Additional features could be used to indicate if the shortest path between two protein pairs adheres to a specific semantic rule covered by the C_{DC}-constraint (see Subsection 5.3.3). For example, a feature might indicate if the shortest path fulfills the scaffold defined for interaction verbs. This can be implemented for all the rules described in Subsection 5.3.3. Another source of meaningful features are patterns introduced in Blaschke *et al.* (1999), Ono *et al.* (2001), and Baumgartner *et al.* (2008). We believe that such features provide semantic clues and will therefore lead to improved robustness of a classifier.

Generalizers modifying node or edge labels (*e.g.,* replacement of interaction words, stemming, and unifying dependency types) are comparably easy to incorporate into a machine learning setting. For instance, token features could be replaced by the respective word stem or lemma when generating the feature representation. It has been shown by Buyko *et al.* (2009), that modifying the dependency tree by trimming dependencies can improve performance for event extraction. It would therefore be interesting to evaluate the impact of G_{CD}, which modifies the dependency tree by removing irrelevant dependency types and attached nodes, in a machine learning setting.

7.2.2 Frequent Subgraph Mining

The shortest path assumption is one of the most widely used concepts in biomedical relationship extraction. In Section 5.4 we observed that shortest path patterns learned on the training corpus achieve a comparably low precision ($\leq 55\,\%$). This indicates that the shortest path assumption alone is insufficient to derive high quality patterns. An alternative idea to the shortest path assumption is to collect frequent subgraphs encompassing the entity pair in question. As the number of annotated PPI corpora is rather small this approach could be used in conjunction with distant supervision. For named entity recognition, a similar approach has been used to collect a large set of surface patterns (Caporaso *et al.*, 2007a; Thomas and Leser, 2013). After manual annotation of the most frequent patterns these approaches achieve excellent results on individual test sets.

7.2.3 Discriminative Pattern Mining

In Chapter 5 we learned patterns from a distantly labeled corpus. So far, patterns are only extracted from positively labeled instances, neglecting the negative instances derived by the closed world assumption. The originally extracted patterns (without refinement) achieve comparably low precision. In other words these patterns provide little discriminative power between the two classes. Discriminative pattern mining (DPM) could be used to overcome this problem (Liu *et al.*, 2014). DPM builds patterns for positive and negative instances. In a second step, DSM searches for patterns with disproportionate frequencies between the two classes and evaluates them by some criterion (*e.g.,* χ^2 or mutual information).

7.2.4 Co-training

In Chapter 4 we discussed the application of self-training for domain adaptation. Self-training applies a model on a large set of unannotated data and uses the most confidently classified data-points to train a new model. However, instances distant from the separating hyperplane will often not end up as support vectors in the next training phase.

An alternative approach for selecting informative instances is co-training (Blum and Mitchell, 1998). Co-training uses two classifiers optimally implementing independent views on the data. Each classifier is trained on the training set and subsequently applied to unlabeled instances (*e.g.,* protein pairs contained in MEDLINE). The instances, most confidently classified by the first classifier are than used as additional instances for the second classifier and vice versa. Thus, co-training potentially chooses informative instances, which are more likely to end up as support vectors than instances selected by self-training. In comparison to self-training, co-training potentially achieves a higher robustness due to the improved instance selection strategy.

In Thomas *et al.* (2012b), we randomly sampled 200,000 co-occurring protein pairs from MEDLINE abstracts and classified them using different classifiers. Figure 7.1 shows a scatter plot for the confidence values between APG and SL predictions on the 200,000 instances. Both classifiers agree on the predicted class label on instances contained in the first and third quadrant (86.9 % of all instances). Whereas the two methods have conflicting results for instances in the second and fourth quadrant. Although there is a correlation between APG and SL predictions (Pearson correlation = 0.60, p-value of $2.3 \cdot 10^{-31}$), we can see that there are several instances confidently classified by only one classifier. These instances should be highly informative for the other classifier and are likely to end up as support vectors when implementing a co-training approach.

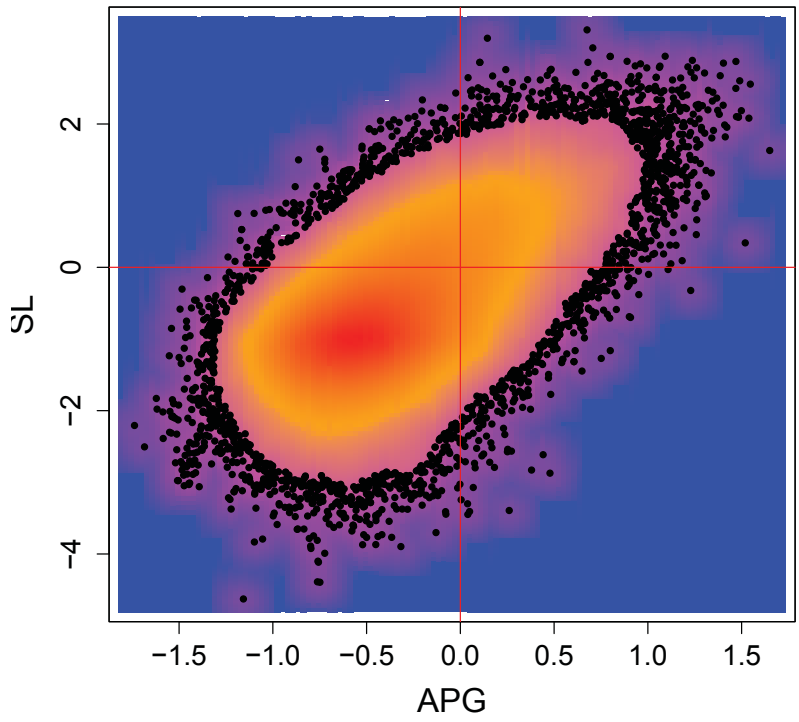


Figure 7.1: Scatter plot for distance to the hyperplane between APG and SL on 200,000 randomly sampled protein pairs from MEDLINE. Warm regions (red) indicate an accumulation of instances whereas cold (blue) regions contain no instances. The 2,000 points in areas with lowest regional density (outliers) are plotted separately.

Bibliography

Airola, A., Pyysalo, S., Björne, J., Pahikkala, T., Ginter, F., and Salakoski, T. (2008). All-paths graph kernel for protein-protein interaction extraction with evaluation of cross-corpus learning. *BMC Bioinformatics*, **9** Suppl 11, S2. (See pp. 30, 33, 35, 39, 49, 65, and 103)

Alex, B., Nissim, M., and Grover, C. (2006). The Impact of Annotation on the Performance of Protein Tagging in Biomedical Text. In *Proceedings of the 5th International Conference on Language Resources and Evaluation*. (See p. 66)

Alfarano, C., Andrade, C. E., Anthony, K., Bahroos, N., Bajec, M., Bantoft, K., Betel, D., Bobechko, B., Boutilier, K., Burgess, E., Buzadzija, K., Cavero, R., D'Abreo, C., Donaldson, I., Dorairajoo, D., Dumontier, M. J., Dumontier, M. R., Earles, V., Farrall, R., Feldman, H., Garderman, E., Gong, Y., Gonzaga, R., Grytsan, V., Gryz, E., Gu, V., Haldorsen, E., Halupa, A., Haw, R., Hrvojic, A., Hurrell, L., Isserlin, R., Jack, F., Juma, F., Khan, A., Kon, T., Konopinsky, S., Le, V., Lee, E., Ling, S., Magidin, M., Moniakis, J., Montojo, J., Moore, S., Muskat, B., Ng, I., Paraiso, J. P., Parker, B., Pintilie, G., Pirone, R., Salama, J. J., Sgro, S., Shan, T., Shu, Y., Siew, J., Skinner, D., Snyder, K., Stasiuk, R., Strumpf, D., Tuekam, B., Tao, S., Wang, Z., White, M., Willis, R., Wolting, C., Wong, S., Wrong, A., Xin, C., Yao, R., Yates, B., Zhang, S., Zheng, K., Pawson, T., Ouellette, B. F. F., and Hogue, C. W. V. (2005). The Biomolecular Interaction Network Database and related tools 2005 update. *Nucleic Acids Research*, **3**3, D418–D424. (See p. 2)

Aranda, B., Achuthan, P., Alam-Faruque, Y., Armean, I., Bridge, A., Derow, C., Feuermann, M., Ghanbarian, A. T., Kerrien, S., Khadake, J., Kerssemakers, J., Leroy, C., Menden, M., Michaut, M., Montecchi-Palazzi, L., Neuhauser, S. N., Orchard, S., Perreau, V., Roechert, B., van Eijk, K., and Hermjakob, H. (2010). The IntAct molecular interaction database in 2010. *Nucleic Acids Research*, **3**8, 525–531. (See pp. 1 and 82)

Arighi, C. N., Lu, Z., Krallinger, M., Cohen, K. B., Wilbur, W. J., Valencia, A., Hirschman, L., and Wu, C. H. (2011). Overview of the BioCreative III Workshop. *BMC Bioinformatics*, **1**2 Suppl 8, S1. (See p. 40)

Aronson, A. R. and Lang, F.-M. (2010). An overview of MetaMap: historical perspective and recent advances. *Journal of the American Medical Informatics Association*, **17**(3), 229–236. (See p. 61)

Arzt, S., Starlinger, J., Arnold, O., Stefan Kröger, S. J., and Leser, U. (2011). PiPa: Custom Integration of Protein Interactions and Pathways. In *41. Jahrestagung der Gesellschaft für Informatik*, Bonn, Germany. (See p. 132)

Bader, G. D., Betel, D., and Hogue, C. W. (2003). BIND: the Biomolecular Interaction Network Database. *Nucleic Acids Research*, **3**1(1), 248–250. (See p. 2)

Bailey, D. G., Dresser, G., and Arnold, J. M. O. (2013). Grapefruit–medication interactions: Forbidden fruit or avoidable consequences? *CMAJ*, **1**85(4), 309–316. (See p. 45)

Balke, W.-T. (2012). Introduction to Information Extraction: Basic Notions and Current Trends. *Datenbank-Spektrum*, **1**2(2), 81–88. (See p. 23)

Ballardini, R., Benevento, M., Arrigoni, G., Pattini, L., and Roda, A. (2011). MassUntangler: a novel alignment tool for label-free liquid chromatography-mass spectrometry proteomic data. *Journal of Chromatography A*, **1**218(49), 8859–8868. (See p. 38)

Bauer-Mehren, A., Furlong, L. I., and Sanz, F. (2009). Pathway databases and tools for their exploitation: benefits, current limitations and challenges. *Molecular Systems Biology*, **5**, 290. (See p. 134)

Baumgartner, W. A., Cohen, K. B., Fox, L. M., Acquaah-Mensah, G., and Hunter, L. (2007). Manual curation is not sufficient for annotation of genomic databases. *Bioinformatics*, **2**3(13), i41–i48. (See p. 2)

Baumgartner, Jr, W. A., Lu, Z., Johnson, H. L., Caporaso, J. G., Paquette, J., Lindemann, A., White, E. K., Medvedeva, O., Cohen, K. B., and Hunter, L. (2008). Concept recognition for extracting protein interaction relations from biomedical text. *Genome Biology*, **9** Suppl 2, S9. (See pp. 24, 30, 33, and 141)

Ben-Hur, A., Ong, C. S., Sonnenburg, S., Schölkopf, B., and Rätsch, G. (2008). Support Vector Machines and Kernels for Computational Biology. *PLoS Computational Biology*, **4**(10), e1000173. (See p. 12)

Berwouts, S., Morris, M. A., Girodon, E., Schwarz, M., Stuhrmann, M., and Dequeker, E. (2011). Mutation nomenclature in practice: Findings and recommendations from the cystic fibrosis external quality assessment scheme. *Human Mutation*, **3**2(11), 1197–1203. (See p. 128)

Best, D. J. and Gipps, P. G. (1974). Algorithm AS 71: The Upper Tail Probabilities of Kendall's Tau. *Journal of the Royal Statistical Society*, **2**3(1), pp. 98–100. (See pp. 69, 84, and 131)

Bies, A., Kulick, S., and Mandel, M. (2005). Parallel Entity and Treebank Annotation. In *Proceedings of the Workshop on Frontiers in Corpus Annotations II: Pie in the Sky*, pages 21–28, Ann Arbor, MI, USA. (See p. 9)

Björne, J., Heimonen, J., Ginter, F., Airola, A., Pahikkala, T., and Salakoski, T. (2009). Extracting Complex Biological Events with Rich Graph-Based Feature Sets. In *Proceedings of the BioNLP 2009 Workshop Companion Volume for Shared Task*, pages 10–18, Boulder, CO, USA. (See p. 7)

Björne, J., Ginter, F., Pyysalo, S., Tsujii, J., and Salakoski, T. (2010). Complex event extraction at PubMed scale. *Bioinformatics*, **2**6(12), i382–i390. (See p. 138)

Björne, J., Ginter, F., Pyysalo, S., Tsujii, J., and Salakoski, T. (2010). Scaling up Biomedical Event Extraction to the Entire PubMed. In *Proceedings of the 2010 Workshop on Biomedical Natural Language Processing*, pages 28–36, Uppsala, Sweden. (See p. 138)

Björne, J., Heimonen, J., Ginter, F., Airola, A., Pahikkala, T., and Salakoski, T. (2011). Extracting Contextualized Complex Biological Events with Rich Graph-Based Features Sets. *Computational Intelligence*, **27**(4), 541–557. (See pp. 49 and 61)

Björne, J., Ginter, F., and Salakoski, T. (2012). University of Turku in the BioNLP'11 Shared Task. *BMC Bioinformatics*, **13** Suppl 11, S4. (See pp. 40 and 78)

Björne, J., Kaewphan, S., and Salakoski, T. (2013). UTurku: Drug Named Entity Recognition and Drug-Drug Interaction Extraction Using SVM Classification and Domain Knowledge. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 651–659, Atlanta, GA, USA. (See p. 61)

Blaschke, C., Andrade, M. A., Ouzounis, C., and Valencia, A. (1999). Automatic extraction of biological information from scientific text: protein-protein interactions. *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology*, pages 60–67. (See pp. 24 and 141)

Blum, A. and Mitchell, T. (1998). Combining Labeled and Unlabeled Data with Co-training. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, pages 92–100. (See p. 142)

Bobic, T. and Klinger, R. (2013). Committee-based Selection of Weakly Labeled Instances for Learning Relation Extraction. In *Proceedings of the Conference on Intelligent Text Processing and Computational Linguistics*. (See p. 112)

Bobic, T., Fluck, J., and Hofmann-Apitius, M. (2013). SCAI: Extracting drug-drug interactions using a rich feature vector. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 675–683, Atlanta, GA, USA. (See p. 62)

Bokharaeian, B. and Diaz, A. (2013). NIL_UCM: Extracting Drug-Drug interactions from text through combination of sequence and tree kernels. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 644–650, Atlanta, GA, USA. (See p. 61)

Bollacker, K., Evans, C., Paritosh, P., Sturge, T., and Taylor, J. (2008). Freebase: A Collaboratively Created Graph Database For Structuring Human Knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250, New York, NY, USA. (See p. 111)

Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). A Training Algorithm for Optimal Margin Classifiers. In *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, pages 144–152. (See p. 12)

Bozek, K., Relógio, A., Kielbasa, S. M., Heine, M., Dame, C., Kramer, A., and Herzel, H. (2009). Regulation of Clock-Controlled Genes in Mammals. *PLoS One*, **4**(3), e4882. (See p. 134)

Breiman, L. (1996). Bagging Predictors. *Machine Learning*, **2**4(2), 123–140. (See pp. 44 and 84)

Breiman, L. (2001). Random Forests. *Machine Learning*, **4**5(1), 5–32. (See p. 45)

Bui, Q.-C., Katrenko, S., and Sloot, P. M. (2011). A hybrid approach to extract protein-protein interactions. *Bioinformatics*, **2**7(2), 259–265. (See pp. 30, 35, 37, 63, and 140)

Bui, Q.-C., Sloot, P. M. A., van Mulligen, E. M., and Kors, J. A. (2014). A novel feature-based approach to extract drug-drug interactions from biomedical text. *Bioinformatics*. (See p. 63)

Bunescu, R., Ge, R., Kate, R. J., Marcotte, E. M., Mooney, R. J., Ramani, A. K., and Wong, Y. W. (2005). Comparative experiments on learning information extractors for proteins and their interactions. *Artificial Intelligence in Medicine*, **3**3(2), 139–155. (See pp. 26 and 91)

Bunescu, R. C. and Mooney, R. J. (2005a). A Shortest Path Dependency Kernel for Relation Extraction. In *Proceedings of the Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 724–731, Vancouver, Canada. (See p. 32)

Bunescu, R. C. and Mooney, R. J. (2005b). Subsequence Kernels for Relation Extraction. In *Proceedings of Advances in Neural Information Processing Systems*, pages 171–178. (See pp. 23, 30, 31, 32, and 50)

Buyko, E., Wermter, J., Poprat, M., and Hahn, U. (2006). Automatically Adapting an NLP Core Engine to the Biology Domain. In *Proceedings of the Joint BioLINK-Bio-Ontologies Meeting 2006*, pages 65–68, Fortaleza, Brazil. (See pp. 68, 82, and 117)

Buyko, E., Faessler, E., Wermter, J., and Hahn, U. (2009). Event Extraction from Trimmed Dependency Graphs. In *Proceedings of the BioNLP 2009 Workshop Companion Volume for Shared Task*, pages 19–27, Boulder, CO, USA. (See p. 141)

Buyko, E., Beisswanger, E., and Hahn, U. (2012). The extraction of pharmacogenetic and pharmacogenomic relations–a case study using PharmGKB. *Pacific Symposium on Biocomputing*, pages 376–387. (See p. 112)

Caporaso, J. G., Baumgartner, W. A., Randolph, D. A., Cohen, K. B., and Hunter, L. (2007a). MutationFinder: a high-performance system for extracting point mutation mentions from text. *Bioinformatics*, **2**3(14), 1862–1865. (See pp. 118 and 141)

Caporaso, J. G., Baumgartner, W. A., Randolph, D. A., Cohen, K. B., and Hunter, L. (2007b). Rapid pattern development for concept recognition systems: application to point mutations. *Journal of Bioinformatics and Computational Biology*, **5**(6), 1233–1259. (See p. 24)

Cavuto, N. J., Woosley, R. L., and Sale, M. (1996). Pharmacies and Prevention of Potentially Fatal Drug Interactions. *JAMA*, **2**75(14), 1086–1087. (See p. 46)

Charniak, E. and Johnson, M. (2005). Coarse-to-Fine n-Best Parsing and MaxEnt Discriminative Reranking. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 173–180, Ann Arbor, MI, USA. (See p. 48)

Charniak, E., Hendrickson, C., Jacobson, N., and Perkowitz, M. (1993). Equations for Part-of-Speech Tagging. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, pages 784–789. (See p. 9)

Chawla, N., Japkowicz, N., and Kotcz, A. (2004). Editorial: Special Issue on Learning from Imbalanced Data Sets. *ACM SIGKDD Explorations Newsletter*, **6**(1), 1–6. (See p. 83)

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, **1**6(1), 321–357. (See p. 62)

Cheitlin, M. D., Hutter, A. M., Brindis, R. G., Ganz, P., Kaul, S., Russell, R. O., and Zusman, R. M. (1999). Use of sildenafil (Viagra) in patients with cardiovascular disease. *Journal of the American College of Cardiology*, **3**3(1), 273–282. (See p. 45)

Cheng, D., Knox, C., Young, N., Stothard, P., Damaraju, S., and Wishart, D. S. (2008). PolySearch: a web-based text mining system for extracting relationships between human diseases, genes, mutations, drugs and metabolites. *Nucleic Acids Research*, **3**6, W399–W405. (See p. 137)

Choi, S.-P. and Myaeng, S.-H. (2010). Simplicity is Better: Revisiting Single Kernel PPI Extraction. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 206–214, Beijing, China. (See pp. 30 and 36)

Chomsky, N. (1957). *Syntactic Structures*. Mouton Classic. (See p. 10)

Chowdhury, F. M., Lavelli, A., and Moschitti, A. (2011). A Study on Dependency Tree Kernels for Automatic Extraction of Protein-Protein Interaction. In *Proceedings of BioNLP 2011 Workshop*, pages 124–133, Portland, OR, USA. (See pp. 30, 31, and 177)

Chowdhury, M. F. M. and Lavelli, A. (2012a). An Evaluation of the Effect of Automatic Preprocessing on Syntactic Parsing for Biomedical Relation Extraction. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation*, pages 544–551, Istanbul, Turkey. (See pp. 31 and 36)

Chowdhury, M. F. M. and Lavelli, A. (2012b). Combining Tree Structures, Flat Features and Patterns for Biomedical Relation Extraction. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 420–429, Avignon, France. (See pp. 31 and 37)

Chowdhury, M. F. M. and Lavelli, A. (2012c). Impact of Less Skewed Distributions on Efficiency and Effectiveness of Biomedical Relation Extraction. In *Proceedings of COLING 2012: Posters*, pages 205–216, Mumbai, India. (See pp. 31, 37, and 59)

Chowdhury, M. F. M. and Lavelli, A. (2013a). Exploiting the Scope of Negations and Heterogeneous Features for Relation Extraction: A Case Study for Drug-Drug Interaction Extraction. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 765–771, Atlanta, GA, USA. (See p. 59)

Chowdhury, M. F. M. and Lavelli, A. (2013b). FBK-irst : A Multi-Phase Kernel Based Approach for Drug-Drug Interaction Detection and Classification that Exploits Linguistic Information. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 351–355, Atlanta, GA, USA. (See pp. 52, 59, and 140)

Cohen, K. B., Verspoor, K., Johnson, H., Roeder, C., Ogren, P., Baumgartner, W., White, E., and Hunter, L. (2009). High-precision biological event extraction with a concept recognizer. In *Proceedings of the BioNLP 2009 Workshop Companion Volume for Shared Task*, pages 50–58, Boulder, CO, USA. (See p. 63)

Cohen, K. B., Johnson, H. L., Verspoor, K., Roeder, C., and Hunter, L. E. (2010). The structural and content aspects of abstracts versus bodies of full text journal articles are different. *BMC Bioinformatics*, **1**1, 492. (See pp. 58, 123, and 126)

Collins, F. S., Brooks, L. D., and Chakravarti, A. (1998). A DNA polymorphism discovery resource for research on human genetic variation. *Genome Research*, **8**(12), 1229–1231. (See p. 128)

Collins, M. and Duffy, N. (2001). Convolution Kernels for Natural Language. In *Proceedings of Neural Information Processing Systems*, pages 625–632. (See pp. 16, 18, and 49)

Comeau, D. C., Islamaj Dogan, R., Ciccarese, P., Cohen, K. B., Krallinger, M., Leitner, F., Lu, Z., Peng, Y., Rinaldi, F., Torii, M., Valencia, A., Verspoor, K., Wiegers, T. C., Wu, C. H., and Wilbur, W. J. (2013). BioC: a minimalist approach to interoperability for biomedical text processing. *Database (Oxford)*, **2**013, bat064. (See p. 137)

Coulet, A., Garten, Y., Dumontier, M., Altman, R., Musen, M., and Shah, N. (2011). Integration and publication of heterogeneous text-mined relationships on the Semantic Web. *Journal of Biomedical Semantics*, **2**(Suppl 2), S10. (See p. 50)

Craven, M. and Kumlien, J. (1999). Constructing Biological Knowledge Bases by Extracting Information from Text Sources. In *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology*, pages 77–86. (See pp. 79 and 111)

Cristianini, N. and Shawe-Taylor, J. (2003). *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 1 edition. (See p. 12)

Daemmrich, A. (2002). A tale of two experts: thalidomide and political engagement in the United States and West Germany. *Social History of Medicine*, **1**5(1), 137–158. (See p. 128)

De Las Rivas, J. and Fontanillo, C. (2010). Protein–Protein Interactions Essentials: Key Concepts to Building and Analyzing Interactome Networks. *PLoS Computational Biology*, **6**(6), e1000807. (See p. 2)

De Marneffe, M., MacCartney, B., and Manning, C. (2006). Generating Typed Dependency Parses from Phrase Structure Parses. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation*, volume 6, pages 449–454. (See p. 48)

De Marneffe, M.-C. and Manning, C. D. (2008). The Stanford Typed Dependencies Representation. In *Proceedings of the Workshop on Cross-Framework and Cross-Domain Parser Evaluation*, pages 1–8, Manchester, UK. (See p. 10)

Den Dunnen, J. T. and Antonarakis, S. E. (2000). Mutation nomenclature extensions and suggestions to describe complex mutations: a discussion. *Human Mutation*, **1**5(1), 7–12. (See p. 128)

D'Eustachio, P. (2011). Reactome knowledgebase of human biological pathways and processes. *Methods in Molecular Biology*, **6**94, 49–61. (See p. 132)

Dietterich, T. G. (1998). Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms. *Neural Computation*, **1**0, 1895–1923. (See pp. 69 and 70)

Ding, J. and Berleant, D. (2003). Extracting biochemical interactions from MEDLINE using a link grammar parser. In *Proceedings of the 15th IEEE International Conference on Tools with Artificial Intelligence*, pages 467–471. (See p. 24)

Ding, J., Berleant, D., Nettleton, D., and Wurtele, E. (2002). Mining MEDLINE: abstracts, sentences, or phrases? *Pacific Symposium on Biocomputing*, pages 326–337. (See pp. 23 and 26)

Dogan, R. I., Murray, G. C., Névéol, A., and Lu, Z. (2009). Understanding PubMed user search behavior through log analysis. *Database (Oxford)*, page bap018. (See p. 115)

Doms, A. and Schroeder, M. (2005). GoPubMed: exploring PubMed with the Gene Ontology. *Nucleic Acids Research*, **33**(Web Server issue), W783–W786. (See p. 137)

Efron, B. (1979). Bootstrap Methods: Another Look at the Jackknife. *The Annals of Statistics*, **7**(1), 1–26. (See p. 20)

Egan, J. (1975). *Signal Detection Theory and ROC-analysis*. Academic Press. (See p. 19)

Erkan, G., Özgür, A., and Radev, D. R. (2007). Semi-Supervised Classification for Extracting Protein Interaction Sentences using Dependency Parsing. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 228–237, Prague, Czech Republic. (See pp. 30, 32, 77, and 103)

Fayruzov, T., Cock, M., Cornelis, C., and Hoste, V. (2008a). DEEPER: A Full Parsing Based Approach to Protein Relation Extraction. In *Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics*, volume 4973, pages 36–47. Springer Berlin Heidelberg. (See p. 30)

Fayruzov, T., De Cock, M., Cornelis, C., and Hoste, V. (2008b). The Role of Syntactic Features in Protein Interaction Extraction. In *Proceedings of the 2nd International Workshop on Data and Text Mining in Bioinformatics*, pages 61–68. (See p. 30)

Fayruzov, T., De Cock, M., Cornelis, C., and Hoste, V. (2009). Linguistic feature analysis for protein interaction extraction. *BMC Bioinformatics*, **10**(1), 374. (See p. 30)

Fernández, J. M., Hoffmann, R., and Valencia, A. (2007). iHOP web services. *Nucleic Acids Research*, **35**(Web Server issue), W21–W26. (See p. 137)

Fleuren, W. W. M., Verhoeven, S., Frijters, R., Heupers, B., Polman, J., van Schaik, R., de Vlieg, J., and Alkema, W. (2011). CoPub update: CoPub 5.0 a text mining system to answer biological questions. *Nucleic Acids Research*, **39**(Web Server issue), W450–W454. (See p. 23)

French, L., Lane, S., Xu, L., Siu, C., Kwok, C., Chen, Y., Krebs, C., and Pavlidis, P. (2012). Application and evaluation of automated methods to extract neuroanatomical connectivity statements from free text. *Bioinformatics*, **28**(22), 2963–2970. (See p. 119)

Fundel, K., Küffner, R., and Zimmer, R. (2007). RelEx–relation extraction using dependency parse trees. *Bioinformatics*, **23**(3), 365–371. (See pp. 24, 26, 30, 32, 95, 102, and 112)

Gale, W. A., Church, K. W., and Yarowsky, D. (1992). One sense per discourse. In *Proceedings of the Workshop on Speech and Natural Language*, pages 233–237, Stroudsburg, PA, USA. (See p. 122)

Gärtner, T., Flach, P. A., and Wrobel, S. (2003). On Graph Kernels: Hardness Results and Efficient Alternatives. In *Learning Theory and Kernel Machines*, volume 2777, pages 129–143. Springer Berlin Heidelberg. (See p. 34)

Geisser, S. (1975). The predictive sample reuse method with applications. *Journal of the American Statistical Association*, **7**0(350), 320–328. (See p. 20)

Gerner, M., Nenadic, G., and Bergman, C. M. (2010). LINNAEUS: A species name identification system for biomedical literature. *BMC Bioinformatics*, **1**1, 85. (See p. 118)

Giardine, B., Borg, J., Higgs, D. R., Peterson, K. R., Philipsen, S., Maglott, D., Singleton, B. K., Anstee, D. J., Basak, A. N., Clark, B., Costa, F. C., Faustino, P., Fedosyuk, H., Felice, A. E., Francina, A., Galanello, R., Gallivan, M. V. E., Georgitsi, M., Gibbons, R. J., Giordano, P. C., Harteveld, C. L., Hoyer, J. D., Jarvis, M., Joly, P., Kanavakis, E., Kollia, P., Menzel, S., Miller, W., Moradkhani, K., Old, J., Papachatzopoulou, A., Papadakis, M. N., Papadopoulos, P., Pavlovic, S., Perseu, L., Radmilovic, M., Riemer, C., Satta, S., Schrijver, I., Stojiljkovic, M., Thein, S. L., Traeger-Synodinos, J., Tully, R., Wada, T., Waye, J. S., Wiemann, C., Zukic, B., Chui, D. H. K., Wajcman, H., Hardison, R. C., and Patrinos, G. P. (2011). Systematic documentation and analysis of human genetic variation in hemoglobinopathies using the microattribution approach. *Nature Genetics*, **4**3(4), 295–301. (See p. 2)

Giuliano, C., Lavelli, A., and Romano, L. (2006). Exploiting Shallow Linguistic Information for Relation Extraction from Biomedical Literature. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 401–408, Trento, Italy. (See pp. 28, 30, 32, 39, 49, 50, and 103)

Giuse, D. A., Giuse, N. B., and Miller, R. A. (1995). Evaluation of long-term maintenance of a large medical knowledge base. *Journal of the American Medical Informatics Association*, **2**(5), 297–306. (See p. 3)

Grishman, R. and Sundheim, B. (1996). Message Understanding Conference-6: a brief history. In *Proceedings of the 16th conference on Computational linguistics - Volume 1*, pages 466–471, Stroudsburg, PA, USA. (See p. 23)

Gu, Q., Dillon, C. F., and Burt, V. L. (2010). Prescription Drug Use Continues to Increase: U.S. Prescription Drug Data for 2007-2008. *NCHS Data Brief*, (42), 1–8. (See p. 45)

Haddow, B. and Alex, B. (2008). Exploiting Multiply Annotated Corpora in Biomedical Information Extraction Tasks. In *Proceedings of the 6th International Language Resources and Evaluation*, Marrakech, Morocco. (See p. 76)

Haider, S. I., Johnell, K., Thorslund, M., and Fastbom, J. (2007). Trends in polypharmacy and potential drug-drug interactions across educational groups in elderly patients in Sweden for the period 1992 - 2002. *International Journal of Clinical Pharmacology and Therapeutics*, **4**5(12), 643–653. (See p. 45)

Hailu, N., Hunter, L. E., and Cohen, K. B. (2013). UColorado_SOM: Extraction of Drug-Drug Interactions from Biomedical Text using Knowledge-rich and Knowledge-poor Features. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 684–688, Atlanta, GA, USA. (See p. 62)

Hakenberg, J., Leser, U., Kirsch, H., and Rebholz-Schuhmann, D. (2006). Collecting a Large Corpus from all of MEDLINE. In *Proceedings of the 2nd International Symposium on Semantic Mining in Biomedicine*, pages 89–92. (See p. 93)

Hakenberg, J., Leaman, R., Vo, N., Jonnalagadda, S., Sullivan, R., Miller, C., Tari, L., Baral, C., and Gonzalez, G. (2010). Efficient extraction of protein-protein interactions from full-text articles. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, **7**(3), 481–494. (See pp. 9 and 25)

Hakenberg, J., Gerner, M., Haeussler, M., Solt, I., Plake, C., Schroeder, M., Gonzalez, G., Nenadic, G., and Bergman, C. M. (2011). The GNAT library for local and remote gene mention normalization. *Bioinformatics*, **2**7(19), 2769–2771. (See pp. 68, 82, and 117)

Hand, D. J. and Till, R. J. (2001). A Simple Generalisation of the Area Under the ROC Curve for Multiple Class Classification Problems. *Machine Learning*, **4**5(2), 171–186. (See p. 20)

Hellmann, S., Lehmann, J., Auer, S., and Nitzschke, M. (2012). NIF combinator: combining NLP tool output. In *Proceedings of the 18th International Conference on Knowledge Engineering and Knowledge Management*, pages 446–449. (See p. 137)

Herrero-Zazo, M., Segura-Bedmar, I., Martínez, P., and Declerck, T. (2013). The DDI corpus: An annotated corpus with pharmacological substances and drug-drug interactions. *Journal of Biomedical Informatics*, **4**6(5), 914–920. (See p. 46)

Hido, S. and Kashima, H. (2009). A linear-time graph kernel. In *Proceedings of the 2009 Ninth IEEE International Conference on Data Mining*, pages 179–188, Washington, DC, USA. (See p. 35)

Hirschman, L., Yeh, A., Blaschke, C., and Valencia, A. (2005). Overview of BioCreAtIvE: critical assessment of information extraction for biology. *BMC Bioinformatics*, **6** Suppl 1, S1. (See p. 40)

Hoffmann, R., Zhang, C., and Weld, D. S. (2010). Learning 5000 Relational Extractors. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 286–295, Uppsala, Sweden. (See pp. 111 and 136)

Huang, D. W., Sherman, B. T., and Lempicki, R. A. (2009). Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists. *Nucleic Acids Research*, **37**(1), 1–13. (See p. 126)

Hunter, L. and Cohen, K. B. (2006). Biomedical language processing: what's beyond PubMed? *Molecular Cell*, **21**(5), 589–594. (See pp. 1 and 115)

Hunter, L., Lu, Z., Firby, J., Baumgartner, Jr, W. A., Johnson, H. L., Ogren, P. V., and Cohen, K. B. (2008). OpenDMAP: An open source, ontology-driven concept analysis engine, with applications to capturing knowledge regarding protein transport, protein interactions and cell-type-specific gene expression. *BMC Bioinformatics*, **9**, 78. (See p. 33)

Hur, J., Schuyler, A. D., States, D. J., and Feldman, E. L. (2009). SciMiner: web-based literature mining tool for target identification and functional enrichment analysis. *Bioinformatics*, **25**(6), 838–840. (See pp. 23 and 137)

Intxaurrondo, A., Surdeanu, M., de Lacalle, O. L., and Agirre, E. (2013). Removing Noisy Mentions for Distant Supervision. In *Proceedings of the 29th Congreso de la Sociedad Española para el Procesamiento del Lenguaje Natural*, volume 51, pages 41–48. (See p. 112)

Irsoy, O., Yildiz, O. T., and Alpaydin, E. (2012). Design and analysis of classifier learning experiments in bioinformatics: survey and case studies. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, **9**(6), 1663–1675. (See p. 12)

Jacob, C. (2010). *Relevanzranking in Lucene im biomedizinischen Kontext*. Studienarbeit, Humboldt Universität zu Berlin. (See p. 120)

Jimeno-Yepes, A. and Aronson, A. (2011). Self-training and co-training in biomedical word sense disambiguation. In *Proceedings of BioNLP 2011 Workshop*, pages 182–183, Portland, OR, USA. (See p. 68)

Joachims, T. (1999). Transductive Inference for Text Classification using Support Vector Machines. In *Proceedings of the Sixteenth International Conference on Machine Learning*, pages 200–209. (See pp. 32 and 78)

Kabiljo, R., Clegg, A. B., and Shepherd, A. J. (2009). A realistic assessment of methods for extracting gene/protein interactions from free text. *BMC Bioinformatics*, **10**, 233. (See pp. 23, 25, 31, 33, 38, and 177)

Kanehisa, M., Goto, S., Sato, Y., Furumichi, M., and Tanabe, M. (2012). KEGG for integration and interpretation of large-scale molecular data sets. *Nucleic Acids Research*, **40**(1), D109–D114. (See p. 132)

Karamanis, N., Lewin, I., Seal, R., Drysdale, R., and Briscoe, E. (2007). Integrating natural language processing with flybase curation. *Pacific Symposium on Biocomputing*, pages 245–256. (See p. 7)

Katrenko, S. and Adriaans, P. (2007). Learning relations from biomedical corpora using dependency trees. In *Knowledge Discovery and Emergent Complexity in Bioinformatics*, volume 4366, pages 61–80. Springer. (See p. 30)

Katrenko, S., Adriaans, P., and van Someren, M. (2010). Using Local Alignments for Relation Recognition. *Journal of Artificial Intelligence Research*, **3**8(1), 1–48. (See p. 30)

Kaufman, S., Rosset, S., and Perlich, C. (2011). Leakage in Data Mining: Formulation, Detection, and Avoidance. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 556–563, New York, NY, USA. (See p. 40)

Kendall, M. G. (1938). A New Measure of Rank Correlation. *Biometrika*, **3**0, 81–93. (See pp. 84 and 131)

Kim, J.-D., Ohta, T., Pyysalo, S., Kano, Y., and Tsujii, J. (2009). Overview of BioNLP'09 Shared Task on Event Extraction. In *Proceedings of the BioNLP 2009 Workshop Companion Volume for Shared Task*, pages 1–9, Boulder, CO, USA. (See pp. 10, 40, and 43)

Kim, J.-D., Wang, Y., Takagi, T., and Yonezawa, A. (2011a). Overview of Genia Event Task in BioNLP Shared Task 2011. In *Proceedings of BioNLP Shared Task 2011 Workshop*, pages 7–15, Portland, OR, USA. (See pp. 11 and 25)

Kim, S., Yoon, J., Yang, J., and Park, S. (2010). Walk-weighted subsequence kernels for protein-protein interaction extraction. *BMC Bioinformatics*, **1**1, 107. (See pp. 26 and 30)

Kim, S., Kwon, D., Shin, S.-Y., and Wilbur, W. J. (2011b). PIE the search: searching PubMed literature for protein interaction information. *Bioinformatics*. (See p. 137)

Knox, C., Law, V., Jewison, T., Liu, P., Ly, S., Frolkis, A., Pon, A., Banco, K., Mak, C., Neveu, V., Djoumbou, Y., Eisner, R., Guo, A. C., and Wishart, D. S. (2011). Drugbank 3.0: a comprehensive resource for 'omics' research on drugs. *Nucleic Acids Research*, **3**9(Database issue), D1035–D1041. (See p. 45)

Kohavi, R. (1995). A Study of Cross-validation and Bootstrap for Accuracy Estimation and Model Selection. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, volume 2, pages 1137–1143. (See p. 21)

Kolářik, C., Klinger, R., Friedrich, C. M., Hofmann-Apitius, M., and Fluck, J. (2008). Chemical Names: Terminological Resources and Corpora Annotation. In *Workshop on Building and evaluating resources for biomedical text mining*, pages 51–58, Marrakech, Morocco. (See p. 118)

Kolářik, C., Klinger, R., and Hofmann-Apitius, M. (2009). Identification of histone modifications in biomedical text for supporting epigenomic research. *BMC Bioinformatics*, **1**0 Suppl 1, S28. (See p. 118)

Krallinger, M., Morgan, A., Smith, L., Leitner, F., Tanabe, L., Wilbur, J., Hirschman, L., and Valencia, A. (2008). Evaluation of text-mining systems for biology: overview of the second biocreative community challenge. *Genome Biology*, **9** Suppl 2, S1. (See p. 40)

Krallinger, M., Rodriguez-Penagos, C., Tendulkar, A., and Valencia, A. (2009). PLAN2L: a web tool for integrated text mining and literature-based bioentity relation extraction. *Nucleic Acids Research*, **37**, W160–W165. (See p. 137)

Kuboyama, T., Hirata, K., Kashima, H., Aoki-Kinoshita, K. F., and Yasuda, H. (2007). A Spectrum Tree Kernel. *Information and Media Technologies*, **2**(1), 292–299. (See pp. 49 and 103)

Kuncheva, L. I. (2004). *Combining Pattern Classifiers: Methods and Algorithms*. Wiley-Interscience. (See p. 44)

Lafferty, J., McCallum, A., and Pereira, F. (2001). Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 282–289, San Francisco, CA, USA. (See p. 8)

Landeghem, S. V., Abeel, T., Saeys, Y., and de Peer, Y. V. (2010). Discriminative and informative features for biomolecular text mining with ensemble feature selection. *Bioinformatics*, **2**6(18), i554–i560. (See p. 77)

Landeghem, S. V., Björne, J., Wei, C.-H., Hakala, K., Pyysalo, S., Ananiadou, S., Kao, H.-Y., Lu, Z., Salakoski, T., de Peer, Y. V., and Ginter, F. (2013). Large-scale event extraction from literature with multi-level gene normalization. *PLoS One*, **8**(4), e55814. (See p. 138)

Laros, J. F. J., Blavier, A., den Dunnen, J. T., and Taschner, P. E. M. (2011). A formalized description of the standard human variant nomenclature in Extended Backus-Naur Form. *BMC Bioinformatics*, **1**2 Suppl 4, S5. (See p. 118)

Leaman, R., Miller, C., and Gonzalez, G. (2009). Enabling Recognition of Diseases in Biomedical Text with Machine Learning: Corpus and Benchmark. In *Proceedings of the 3rd International Symposium on Languages in Biology and Medicine*, pages 82–89. (See p. 118)

Lease, M. and Charniak, E. (2005). Parsing Biomedical Literature. In *Proceedings of the Second International Joint Conference on Natural Language Processing*, volume 3651, pages 58–69. (See p. 122)

Lee, J., Kim, S., Lee, S., Lee, K., and Kang, J. (2012). High Precision Rule Based PPI Extraction and Per-pair Basis Performance Evaluation. In *Proceedings of the ACM Sixth International Workshop on Data and Text Mining in Biomedical Informatics*, pages 69–76, New York, NY, USA. (See p. 31)

Lehnert, W. G., Cardie, C., Fisher, D., McCarthy, J., Riloff, E., and Soderland, S. (1992). University of Massachusetts: Description of the CIRCUS System as Used for MUC-4. In *Fourth Message Understanding Conference*, pages 282–288. (See p. 25)

Leitner, F., Mardis, S. A., Krallinger, M., Cesareni, G., Hirschman, L. A., and Valencia, A. (2010). An Overview of BioCreative II.5. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, **7**(3), 385–399. (See pp. 25, 40, and 43)

Li, Y., Hu, X., Lin, H., and Yang, Z. (2010). Learning an enriched representation from unlabeled data for protein-protein interaction extraction. *BMC Bioinformatics*, **1**1 Suppl 2, S7. (See p. 30)

Li, Y., Hu, X., Lin, H., and Yang, Z. (2011). A framework for semisupervised feature generation and its applications in biomedical literature mining. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, **8**(2), 294–307. (See pp. 31 and 36)

Liu, B., Qian, L., Wang, H., and Zhou, G. (2010a). Dependency-Driven Feature-based Learning for Extracting Protein-Protein Interactions from Biomedical Text. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 757–765, Beijing, China. (See pp. 30 and 112)

Liu, H., Keselj, V., and Blouin, C. (2010b). Biological event extraction using subgraph matching. In *Proceedings of the 4th International Symposium on Semantic Mining in Biomedicine*, pages 110–115. (See p. 9)

Liu, H., Komandur, R., and Verspoor, K. (2011). From Graphs to Events: A Subgraph Matching Approach for Information Extraction from Biomedical Text. In *Proceedings of BioNLP Shared Task 2011 Workshop*, pages 164–172, Portland, OR, USA. (See p. 112)

Liu, H., Christiansen, T., Baumgartner, W., and Verspoor, K. (2012). Biolemmatizer: a lemmatization tool for morphological processing of biomedical text. *Journal of Biomedical Semantics*, **3**(1), 3. (See pp. 50 and 93)

Liu, H., Hunter, L., Kešelj, V., and Verspoor, K. (2013a). Approximate Subgraph Matching-Based Literature Mining for Biomedical Events and Relations. *PLoS One*, **8**(4), e60954. (See pp. 106 and 108)

Liu, H., Verspoor, K., Comeau, D. C., MacKinlay, A., and Wilbur, W. J. (2013b). Generalizing an Approximate Subgraph Matching-based System to Extract Events in Molecular Biology and Cancer Genetics. In *Proceedings of the BioNLP Shared Task 2013 Workshop*, pages 76–85, Sofia, Bulgaria. (See pp. 112 and 113)

Liu, X., Wu, J., Gu, F., Wang, J., and He, Z. (2014). Discriminative pattern mining and its applications in bioinformatics. *Briefings in Bioinformatics*. (See p. 142)

Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N., and Watkins, C. (2002). Text Classification Using String Kernels. *Journal of Machine Learning Research*, **2**, 419–444. (See p. 31)

Lu, Z. (2011). PubMed and beyond: a survey of web tools for searching biomedical literature. *Database (Oxford)*, **2**011, baq036. (See pp. 115, 117, and 137)

MacKinlay, A., Martinez, D., Jimeno Yepes, A., Liu, H., Wilbur, W. J., and Verspoor, K. (2013). Extracting Biomedical Events and Modifications Using Subgraph Matching with Noisy Training Data. In *Proceedings of the BioNLP Shared Task 2013 Workshop*, pages 35–44, Sofia, Bulgaria. (See pp. 78 and 113)

Maglott, D., Ostell, J., Pruitt, K. D., and Tatusova, T. (2011). Entrez Gene: gene-centered information at NCBI. *Nucleic Acids Research*, **3**9, D52–D57. (See p. 22)

Magrane, M. and Uniprot Consortium (2011). UniProt Knowledgebase: a hub of integrated protein data. *Database (Oxford)*, **2**011. (See p. 1)

Mann, H. B. and Whitney, D. R. (1947). On a Test of Whether one of Two Random Variables is Stochastically Larger than the Other. *Annals of Mathematical Statistics*, **1**8(1), 50–60. (See pp. 52 and 84)

Marcus, M. P., Santorini, B., and Marcinkiewicz, M. A. (1993). Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*, **1**9(2), 313–330. (See pp. 9 and 10)

Mazumder, R., Natale, D. A., Julio, J. A. E., Yeh, L.-S., and Wu, C. H. (2010). Community annotation in biology. *Biology Direct*, **5**, 12. (See p. 2)

McCallum, A., Schultz, K., and Singh, S. (2009). Factorie: Probabilistic programming via imperatively defined factor graphs. In *Proceedings of Advances in Neural Information Processing Systems 22*, pages 1249–1257. (See p. 112)

McClosky, D. (2010). *Any Domain Parsing: Automatic Domain Adaptation for Natural Language Parsing*. Ph.D. thesis, Brown University. (See p. 48)

McClosky, D. and Charniak, E. (2008). Self-Training for Biomedical Parsing. In *Proceedings of the Association for Computational Linguistics*, pages 101–104, Columbus, OH, USA. (See p. 68)

McClosky, D., Charniak, E., and Johnson, M. (2006a). Effective Self-Training for Parsing. In *Proceedings of the Conference on Human Language Technology and North American chapter of the Association for Computational Linguistics*, pages 152–159, Morristown, NJ, USA. (See p. 77)

McClosky, D., Charniak, E., and Johnson, M. (2006b). Reranking and Self-Training for Parser Adaptation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 337–344, Morristown, NJ, USA. (See pp. 67, 68, and 122)

McEntyre, J. R., Ananiadou, S., Andrews, S., Black, W. J., Boulderstone, R., Buttery, P., Chaplin, D., Chevuru, S., Cobley, N., Coleman, L.-A., Davey, P., Gupta, B., Haji-Gholam, L., Hawkins, C., Horne, A., Hubbard, S. J., Kim, J.-H., Lewin, I., Lyte, V., MacIntyre, R., Mansoor, S., Mason, L., McNaught, J., Newbold, E., Nobata, C., Ong, E., Pillai, S., Rebholz-Schuhmann, D., Rosie, H., Rowbotham, R., Rupp, C. J., Stoehr, P., and Vaughan, P. (2011). UKPMC: a full text article resource for the life sciences. *Nucleic Acids Research*, **3**9(Database issue), D58–D65. (See p. 137)

McNemar, Q. (1947). Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, **1**2(2), 153–157. (See pp. 63 and 69)

Messersmith, W. A. and Ahnen, D. J. (2008). Targeting EGFR in colorectal cancer. *New England Journal of Medicine*, **3**59(17), 1834–1836. (See p. 136)

Mintz, M., Bills, S., Snow, R., and Jurafsky, D. (2009). Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, volume 2, pages 1003–1011. (See pp. 79 and 111)

Mitchell, D. W. (2004). 88.27 More on Spreads and Non-Arithmetic Means. *The Mathematical Gazette*, **8**8(511), 142–144. (See p. 19)

Mitsumori, T., Murata, M., Fukuda, Y., Doi, K., and Doi, H. (2006). Extracting Protein-Protein Interaction Information from Biomedical Text with SVM. *IEICE Transactions on Information and Systems*, **E**89-D, 2464–2466. (See p. 30)

Miwa, M., Sætre, R., Miyao, Y., and Tsujii, J. (2009a). A Rich Feature Vector for Protein-Protein Interaction Extraction from Multiple Corpora. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 121–130, Singapore. (See pp. 30, 34, 38, 58, and 77)

Miwa, M., Sætre, R., Miyao, Y., and Tsujii, J. (2009b). Protein-protein interaction extraction by leveraging multiple kernels and parsers. *International Journal of Medical Informatics*, **7**8(12), e39–e46. (See pp. 30, 34, and 35)

Miwa, M., Sætre, R., Miyao, Y., and Tsujii, J. (2010). Entity-Focused Sentence Simplification for Relation Extraction. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 788–796, Beijing, China. (See pp. 30, 35, and 65)

Miyao, Y., Sætre, R., Sagae, K., Matsuzaki, T., and Tsujii, J. (2008). Task-oriented Evaluation of Syntactic Parsers and Their Representations. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*, pages 46–54, Columbus, OH, USA. (See pp. 30 and 38)

Morgan, A. A., Lu, Z., Wang, X., Cohen, A. M., Fluck, J., Ruch, P., Divoli, A., Fundel, K., Leaman, R., Hakenberg, J., Sun, C., hui Liu, H., Torres, R., Krauthammer, M., Lau, W. W., Liu, H., Hsu, C.-N., Schuemie, M., Cohen, K. B., and Hirschman, L.

(2008). Overview of BioCreative II gene normalization. *Genome Biology*, **9**, S3. (See p. 117)

Moschitti, A. (2006). Efficient Convolution Kernels for Dependency and Constituent Syntactic Trees. In *Proceedings of the 17th European conference on Machine Learning*, pages 318–329. (See pp. 16, 18, and 49)

Nakagawa, T. (2004). Chinese and Japanese Word Segmentation Using Word-Level and Character-Level Information. In *Proceedings of the 20th International Conference on Computational Linguistics*, pages 466–472, Geneva, Switzerland. (See p. 9)

Nan, Y., Chai, K. M. A., Lee, W. S., and Chieu, H. L. (2012). Optimizing F-measure: A Tale of Two Approaches. In *Proceedings of the 29th International Conference on Machine Learning*, pages 289–296. (See p. 57)

Nédellec, C. (2005). Learning language in logic-genic interaction extraction challenge. In *Proceedings of the Learning Language in Logic 2005 Workshop at the International Conference on Machine Learning*, volume 18, pages 97–99, Bonn, Germany. (See p. 26)

Nédellec, C., Bossy, R., Kim, J.-D., Kim, J.-J., Ohta, T., Pyysalo, S., and Zweigenbaum, P. (2013). Overview of BioNLP Shared Task 2013. In *Proceedings of the BioNLP Shared Task 2013 Workshop*, pages 1–7, Sofia, Bulgaria. (See pp. 25 and 40)

Neves, M., Carazo, J.-M., and Pascual-Montano, A. (2009). Extraction of biomedical events using case-based reasoning. In *Proceedings of the BioNLP 2009 Workshop Companion Volume for Shared Task*, pages 68–76, Boulder, CO, USA. (See p. 49)

Ng, A. Y. (1997). Preventing Overfitting of Cross-Validation Data. In *Proceedings of the Fourteenth International Conference on Machine Learning*, pages 245–253. (See p. 21)

Nguyen, Q. L., Tikk, D., and Leser, U. (2010). Simple tricks for improving pattern-based information extraction from the biomedical literature. *Journal of Biomedical Semantics*, **1**(1), 9. (See p. 103)

Nguyen, T. V. T. and Moschitti, A. (2011). End-to-End Relation Extraction Using Distant Supervision from External Semantic Repositories. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 277–282, Portland, OR, USA. (See p. 111)

Nguyen, T.-V. T., Moschitti, A., and Riccardi, G. (2009). Convolution Kernels on Constituent, Dependency and Sequential Structures for Relation Extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1378–1387, Singapore. (See p. 30)

Niu, Y., Otasek, D., and Jurisica, I. (2010). Evaluation of linguistic features useful in extraction of interactions from PubMed; application to annotating known, high-throughput and predicted interactions in I2D. *Bioinformatics*, **2**6(1), 111–119. (See pp. 30 and 35)

Nivre, J. (2005). Dependency grammar and dependency parsing. Technical report, Växjö University. (See p. 11)

Ogino, S. and Wilson, R. B. (2004). Importance of standard nomenclature for SMN1 small intragenic ("subtle") mutations. *Human Mutation*, **2**3(4), 392–393. (See pp. 115 and 128)

Ono, T., Hishigaki, H., Tanigami, A., and Takagi, T. (2001). Automated extraction of information on protein-protein interactions from the biological literature. *Bioinformatics*, **1**7(2), 155–161. (See pp. 24 and 141)

Palaga, P. (2009). *Extracting Relations from Biomedical Texts Using Syntactic Information*. Master's Thesis, Technische Universität Berlin. (See pp. 30 and 49)

Palidwor, G. A. and Andrade-Navarro, M. A. (2010). MLTrends: Graphing MEDLINE term usage over time. *Journal of Biomedical Discovery and Collaboration*, **5**, 1–6. (See p. 126)

Pan, S. J. and Yang, Q. (2010). A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering*, **2**2(10), 1345–1359. (See p. 66)

Pantel, P. and Lin, D. (2002). Discovering Word Senses from Text. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 613–619, New York, NY, USA. (See p. 113)

Pearson, K. (1900). On the Criterion that a Given System of Deviations from the Probable in the Case of a Correlated System of Variables is Such that it can be Reasonably Supposed to Have Arisen from Random Sampling. *Philosophical Magazine Series 5*, **5**0(302), 157–175. (See p. 131)

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, **1**2, 2825–2830. (See pp. 13 and 173)

Pfeiffer, T. and Hoffmann, R. (2007). Temporal patterns of genes in scientific publications. *Proceedings of the National Academy of Sciences of the United States of America*, **1**04(29), 12052–12056. (See p. 126)

Pietschmann, S. (2009). *Relationsextraktion durch Frequent Pattern in Dependency Graphen*. Master's thesis, Humboldt Universität zu Berlin. (See p. 91)

Plake, C., Schiemann, T., Pankalla, M., Hakenberg, J., and Leser, U. (2006). AliBaba: PubMed as a graph. *Bioinformatics*, **2**2(19), 2444–2445. (See pp. 118 and 137)

Polikar, R. (2006). Ensemble Based Systems in Decision Making. *IEEE Circuits and Systems Magazine*, **6**(3), 21–45. (See p. 43)

Porter, M. F. (1980). An algorithm for suffix stripping. *Program: Electronic Library & Information Systems*, **4**0(3), 211–218. (See p. 93)

Proux, D., Rechenmann, F., Julliard, L., Pillet, V., and Jacq, B. (1998). Detecting Gene Symbols and Names in Biological Texts: A First Step toward Pertinent Information Extraction. *Genome informatics. Workshop on Genome Informatics*, **9**, 72–80. (See p. 21)

Pyysalo, S., Ginter, F., Heimonen, J., Björne, J., Boberg, J., Järvinen, J., and Salakoski, T. (2007). BioInfer: a corpus for information extraction in the biomedical domain. *BMC Bioinformatics*, **8**, 50. (See p. 26)

Pyysalo, S., Airola, A., Heimonen, J., Björne, J., Ginter, F., and Salakoski, T. (2008a). Comparative analysis of five protein-protein interaction corpora. *BMC Bioinformatics*, **9** Suppl 3, S6. (See pp. 23, 26, 28, 31, 33, 39, 65, 102, 103, 112, and 177)

Pyysalo, S., Sætre, R., Tsujii, J., and Salakoski, T. (2008b). Why Biomedical Relation Extraction Results are Incomparable and What to do about it. In *Proceedings of the Third International Symposium on Semantic Mining in Biomedicine*, pages 149–152. (See p. 27)

Qian, L. and Zhou, G. (2012). Tree kernel-based protein-protein interaction extraction from biomedical literature. *Journal of Biomedical Informatics*, **4**5(3), 535–543. (See pp. 31 and 36)

Raja, K., Subramani, S., and Natarajan, J. (2013). PPInterFinder–a mining tool for extracting causal relations on human proteins from literature. *Database (Oxford)*, **2**013, bas052. (See p. 31)

Rastegar-Mojarad, M., Boyce, R. D., and Prasad, R. (2013). UWM-TRIADS: Classifying Drug-Drug Interactions with Two-Stage SVM and Post-Processing. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 667–674, Atlanta, GA, USA. (See p. 62)

Ravikumar, K. E., Liu, H., Cohn, J., Wall, M., and Verspoor, K. (2011). Pattern Learning through Distant Supervision for Extraction of Protein-Residue Associations in the Biomedical Literature. In *Proceedings of the 2011 10th International Conference on Machine Learning and Applications and Workshops - Volume 02*, volume 2, pages 59–65. (See pp. 112 and 113)

Rebholz-Schuhmann, D., Kirsch, H., Arregui, M., Gaudan, S., Riethoven, M., and Stoehr, P. (2007). EBIMed–text crunching to gather facts for proteins from Medline. *Bioinformatics*, **2**3(2), e237–e244. (See pp. 23 and 137)

Rebholz-Schuhmann, D., Arregui, M., Gaudan, S., Kirsch, H., and Jimeno, A. (2008). Text processing through Web services: calling Whatizit. *Bioinformatics*, **2**4(2), 296–298. (See p. 137)

Rehman, W., Arfons, L. M., and Lazarus, H. M. (2011). The Rise, Fall and Subsequent Triumph of Thalidomide: Lessons Learned in Drug Development. *Therapeutic Advances in Hematology*, **2**(5), 291–308. (See p. 128)

Reichart, R. and Rappoport, A. (2007). Self-Training for Enhancement and Domain Adaptation of Statistical Parsers Trained on Small Datasets. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 616–623, Prague, Czech Republic. (See pp. 68 and 77)

Relógio, A., Thomas, P., Medina-Pérez, P., Reischl, S., Bervoets, S., Gloc, E., Riemer, P., Mang-Fatehi, S., Maier, B., Schäfer, R., Leser, U., Herzel, H., Kramer, A., and Sers, C. (2014). Ras-mediated deregulation of the circadian clock in cancer. *PLoS Genetics*, **1**0(5), e1004338. (See p. 140)

Riedel, S., Yao, L., and McCallum, A. (2010). Modeling Relations and Their Mentions without Labeled Text . In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 148–163. (See p. 112)

Riesbeck, C. (1986). From Conceptual Analyzer to Direct Memory Access Parsing: An Overview. In *Advances in Cognitive Science*, pages 236–258. (See p. 33)

Riley, M. D. (1989). Some applications of tree-based modelling to speech and language. In *Proceedings of the Workshop on Speech and Natural Language*, pages 339–352, Stroudsburg, PA, USA. (See p. 8)

Rinaldi, F., Schneider, G., Kaljurand, K., Hess, M., and Romacker, M. (2006). An environment for relation mining over richly annotated corpora: the case of GENIA. *BMC Bioinformatics*, **7** Suppl 3, S3. (See p. 24)

Rocktäschel, T., Weidlich, M., and Leser, U. (2012). ChemSpot: A Hybrid System for Chemical Named Entity Recognition. *Bioinformatics*, **2**8(12), 1633–1640. (See p. 118)

Rodriguez-Esteban, R. (2009). Biomedical text mining and its applications. *PLoS Computational Biology*, **5**(12), e1000597. (See p. 137)

Rodriguez-Esteban, R. and Loging, W. T. (2013). Quantifying the complexity of medical research. *Bioinformatics*. (See p. 126)

Rodriguez-Penagos, C., Salgado, H., Martinez-Flores, I., and Collado-Vides, J. (2007). Automatic reconstruction of a bacterial regulatory network using Natural Language Processing. *BMC Bioinformatics*, **8**, 293. (See p. 132)

Sætre, R., Sagae, K., and Tsujii, J. (2007). Syntactic features for protein-protein interaction extraction. In *Proceedings of the 2nd International Symposium on Languages in Biology and Medicine*, volume 319, pages 6.1–6.14. (See pp. 28, 30, 31, 33, 38, and 177)

Sætre, R., Yoshida, K., Miwa, M., Matsuzaki, T., Kano, Y., and Tsujii, J. (2010). Extracting Protein Interactions from Text with the Unified AkaneRE Event Extraction System. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, **7**(3), 442–453. (See p. 25)

Salzberg, S. (1997). On Comparing Classifiers: Pitfalls to Avoid and a Recommended Approach. *Data Mining and Knowledge Discovery*, **1**(3), 317–328. (See p. 21)

Sanchez-Cisneros, D. (2013). UC3M: A kernel-based approach to identify and classify DDIs in bio-medical texts. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 617–621, Atlanta, GA, USA. (See p. 61)

Sayers, E. W., Barrett, T., Benson, D. A., Bolton, E., Bryant, S. H., Canese, K., Chetvernin, V., Church, D. M., Dicuccio, M., Federhen, S., Feolo, M., Fingerman, I. M., Geer, L. Y., Helmberg, W., Kapustin, Y., Krasnov, S., Landsman, D., Lipman, D. J., Lu, Z., Madden, T. L., Madej, T., Maglott, D. R., Marchler-Bauer, A., Miller, V., Karsch-Mizrachi, I., Ostell, J., Panchenko, A., Phan, L., Pruitt, K. D., Schuler, G. D., Sequeira, E., Sherry, S. T., Shumway, M., Sirotkin, K., Slotta, D., Souvorov, A., Starchenko, G., Tatusova, T. A., Wagner, L., Wang, Y., Wilbur, W. J., Yaschenko, E., and Ye, J. (2012). Database resources of the National Center for Biotechnology Information. *Nucleic Acids Research*, **4**0(1), D13–D25. (See p. 134)

Schuemie, M. J., Weeber, M., Schijvenaars, B. J. A., van Mulligen, E. M., van der Eijk, C. C., Jelier, R., Mons, B., and Kors, J. A. (2004). Distribution of information in biomedical abstracts and full-text publications. *Bioinformatics*, **2**0(16), 2597–2604. (See p. 115)

Schwartz, A. S. and Hearst, M. A. (2003). A Simple Algorithm for Identifying Abbreviation Definitions in Biomedical Text. In *Proceedings of the Pacific Symposium on Biocomputing*, pages 451–462. (See p. 117)

Schweikert, G., Widmer, C., Schölkopf, B., and Rätsch, G. (2008). An Empirical Analysis of Domain Adaptation Algorithms for Genomic Sequence Analysis. In *Proceedings of Advances in Neural Information Processing Systems*, pages 1433–1440. (See p. 34)

Segura-Bedmar, I., Martínez, P., and Sanchez-Cisneros, D., editors (2011a). *Proceedings of the 1st Challenge Task on Drug-Drug Interaction Extraction*. (See p. 40)

Segura-Bedmar, I., Martínez, P., and Sanchez-Cisneros, D. (2011b). The 1st DDIExtraction-2011 Challenge Task: Extraction of Drug-Drug Interactions from Biomedical Text. In *Proceedings of the 1st Challenge Task on Drug-Drug Interaction Extraction 2011*, pages 1–9, Huelva, Spain. (See p. 46)

Segura-Bedmar, I., Martínez, P., and Herrero Zazo, M. (2013). SemEval-2013 Task 9 : Extraction of Drug-Drug Interactions from Biomedical Texts (DDIExtraction 2013). In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2:*

Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013), pages 341–350, Atlanta, GA, USA. (See pp. 25, 40, 43, and 56)

Segura-Bedmar, I., Martínez, P., and Herrero-Zazo, M. (2014). Lessons learnt from the DDIExtraction-2013 Shared Task. *Journal of Biomedical Informatics*. (See p. 63)

Seringhaus, M. R. and Gerstein, M. B. (2007). Publishing perishing? Towards tomorrow's information architecture. *BMC Bioinformatics*, **8**, 17. (See p. 2)

Shannon, C. E. (1948). A mathematical theory of communication. *Bell system technical journal*, **27**(3), 379–423. (See p. 52)

Simões, G., Galhardas, H., and Matos, D. (2013). A Labeled Graph Kernel for Relationship Extraction. *CoRR*. (See p. 31)

Sing, T., Sander, O., Beerenwinkel, N., and Lengauer, T. (2005). ROCR: visualizing classifier performance in R. *Bioinformatics*, **21**(20), 3940–3941. (See pp. 20 and 173)

Smalley, W., Shatin, D., Wysowski, D. K., Gurwitz, J., Andrade, S. E., Goodman, M., Chan, K. A., Platt, R., Schech, S. D., and Ray, W. A. (2000). Contraindicated Use of CisaprideImpact of Food and Drug Administration Regulatory Action. *JAMA*, **28**4(23), 3036–3039. (See p. 46)

Smialowski, P., Pagel, P., Wong, P., Brauner, B., Dunger, I., Fobo, G., Frishman, G., Montrone, C., Rattei, T., Frishman, D., and Ruepp, A. (2010). The Negatome database: a reference set of non-interacting protein pairs. *Nucleic Acids Research*, **3**8, D540–D544. (See p. 83)

Smith, L., Rindflesch, T., and Wilbur, W. J. (2004). MedPost: a part-of-speech tagger for bioMedical text. *Bioinformatics*, **2**0(14), 2320–2321. (See p. 9)

Solt, I., Szidarovszky, F. P., and Tikk, D. (2010). Concept, assertion and relation extraction at the 2010 i2b2 relation extraction challenge using parsing information and dictionaries. In *Proceedings of the 2010 I2B2/VA Workshop on Challenges in Natural Language*, pages 1–6, Washington, DC, USA. (See pp. 49 and 119)

Spasic, I., Ananiadou, S., and Tsujii, J. (2005). MaSTerClass: a case-based reasoning system for the classification of biomedical terms. *Bioinformatics*, **21**(11), 2748–2758. (See p. 50)

Stamatatos, E., Fakotakis, N., and Kokkinakis, G. (1999). Automatic extraction of rules for sentence boundary disambiguation. In *Proceedings of the Workshop on Machine Learning in Human Language Technology ECCAI Advanced Course on Artificial Intelligence*, pages 88–92, Chania, Greece. (See p. 8)

Stenetorp, P., Pyysalo, S., Topić, G., Ohta, T., Ananiadou, S., and Tsujii, J. (2012). BRAT: a Web-based Tool for NLP-Assisted Text Annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 102–107, Avignon, France. (See pp. 26 and 173)

Strötgen, J., Fluck, J., and Holler, A. (2009). Dependenz-basierte Relationsextraktion mit der UIMA-basierten Textmining Pipeline UTEMPL. In *Proceedings of the Biennial GSCL Conference 2009*, pages 125–136. (See p. 30)

Surdeanu, M., Tibshirani, J., Nallapati, R., and Manning, C. D. (2012). Multi-instance Multi-label Learning for Relation Extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 455–465, Stroudsburg, PA, USA. (See p. 111)

Swampillai, K. and Stevenson, M. (2010). Inter-sentential Relations in Information Extraction Corpora. In *Proceedings of the Seventh conference on International Language Resources and Evaluation*, Valletta, Malta. (See p. 7)

Takahashi, J. S., Hong, H.-K., Ko, C. H., and McDearmon, E. L. (2008). The genetics of mammalian circadian order and disorder: implications for physiology and disease. *Nature Reviews Genetics*, **9**, 764–775. (See p. 134)

Tamames, J. and Valencia, A. (2006). The success (or not) of HUGO nomenclature. *Genome Biology*, **7**(5), 402. (See p. 21)

Temkin, J. and Gilder, M. (2003). Extraction of protein interaction information from unstructured text using a context-free grammar. *Bioinformatics*, **1**9(16), 2046–2053. (See p. 93)

Tesnière, L. (1959). *Éléments de syntaxe structurale*. Klincksieck, Paris. (See pp. 10 and 11)

Thomas, P. and Leser, U. (2013). HistoNer: Histone modification extraction from text. In *Proceedings of BioLINK Special Interest Group*, pages 52–55. (See pp. 24, 118, and 141)

Thomas, P., Starlinger, J., Jacob, C., Solt, I., Hakenberg, J., and Leser, U. (2010). GeneView – Gene Centric Ranking of Biomedical Text. In *Proceedings of the BioCreative III Workshop*, pages 137–142, Bethesda, MD, USA. (See p. 5)

Thomas, P., Klinger, R., Furlong, L. I., Hofmann-Apitius, M., and Friedrich, C. M. (2011a). Challenges in the association of human single nucleotide polymorphism mentions with unique database identifiers. *BMC Bioinformatics*, **1**2 Suppl 4, S4. (See p. 113)

Thomas, P., Solt, I., Klinger, R., and Leser, U. (2011b). Learning Protein Protein Interaction Extraction using Distant Supervision. In *Proceedings of Workshop on Robust Unsupervised and Semisupervised Methods in Natural Language Processing*, pages 25–32, Hissar, Bulgaria. (See p. 5)

Thomas, P., Pietschmann, S., Solt, I., Tikk, D., and Leser, U. (2011c). Not all links are equal: Exploiting Dependency Types for the Extraction of Protein-Protein Interactions from Text. In *Proceedings of BioNLP 2011 Workshop*, pages 1–9, Portland, OR, USA. (See p. 5)

Thomas, P., Neves, M., Solt, I., Tikk, D., and Leser, U. (2011d). Relation Extraction for Drug-Drug Interactions using Ensemble Learning. In *Proceedings of the 1st Challenge Task on Drug-Drug Interaction Extraction 2011*, pages 11–18. (See pp. 48, 49, and 118)

Thomas, P., Starlinger, J., Vowinkel, A., Arzt, S., and Leser, U. (2012a). GeneView: a comprehensive semantic search engine for PubMed. *Nucleic Acids Research*, **4**0(Web Server issue), W585–W591. (See p. 5)

Thomas, P., Bobic, T., Leser, U., Hofmann-Apitius, M., and Klinger, R. (2012b). Weakly Labeled Corpora as Silver Standard for Drug-Drug and Protein-Protein Interaction. In *Proceedings of the Workshop on Building and Evaluating Resources for Biomedical Text Mining (BioTxtM) on Language Resources and Evaluation Conference (LREC)*, pages 63–70, Instanbul, Turkey. (See pp. 81, 112, and 142)

Thomas, P., Starlinger, J., and Leser, U. (2013a). Experiences from Developing the Domain-Specific Entity Search Engine GeneView. In *In proceedings of Datenbanksysteme für Business, Technologie und Web*, pages 225–239. (See p. 5)

Thomas, P., Neves, M., Rocktäschel, T., and Leser, U. (2013b). WBI-DDI: Drug-Drug Interaction Extraction using Majority Voting. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 628–635, Atlanta, GA, USA. (See p. 4)

Thomas, P., Durek, P., Solt, I., Klinger, B., Witzel, F., Schulthess, P., Mayer, Y., Tikk, D., Blüthgen, N., and Leser, U. (2014a). Computer-assisted curation of a human regulatory core network from the biological literature. *Bioinformatics*. (See p. 140)

Thomas, P., Rocktäschel, T., Mayer, Y., and Leser, U. (2014b). SETH: SNP Extraction Tool for Human Variations. `http://rockt.github.io/SETH/`. (See p. 118)

Tian, Y., McEachin, R. C., Santos, C., States, D. J., and Patel, J. M. (2007). SAGA: a subgraph matching tool for biological graphs. *Bioinformatics*, **2**3(2), 232–239. (See p. 108)

Tikk, D., Thomas, P., Palaga, P., Hakenberg, J., and Leser, U. (2010). A comprehensive benchmark of kernel methods to extract protein-protein interactions from literature. *PLoS Computational Biology*, **6**, e1000837. (See pp. 25, 32, 37, 38, 39, 49, 52, 65, 66, 67, 72, 78, 84, 102, 103, 118, 174, and 178)

Tikk, D., Solt, I., Thomas, P., and Leser, U. (2013). A detailed error analysis of 13 kernel methods for protein-protein interaction extraction. *BMC Bioinformatics*, **1**4, 12. (See pp. 25, 31, 39, and 103)

Tomanek, K., Wermter, J., and Hahn, U. (2007). A reappraisal of sentence and token splitting for life sciences documents. *Studies in Health Technology and Informatics*, **1**29(Pt 1), 524–528. (See pp. 8 and 9)

Toutanova, K., Klein, D., Manning, C. D., and Singer, Y. (2003). Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 173–180, Stroudsburg, PA, USA. (See p. 9)

Tsujii, J., Kim, J.-D., and Pyysalo, S., editors (2011). *Proceedings of the BioNLP Shared Task*. (See p. 40)

Tsuruoka, Y., Miwa, M., Hamamoto, K., Tsujii, J., and Ananiadou, S. (2011). Discovering and visualizing indirect associations between biomedical concepts. *Bioinformatics*, **27**(13), i111–i119. (See p. 137)

Turner, B. (2005). Reading signals on the nucleosome with a new nomenclature for modified histones. *Nature Structural and Molecular Biology*, **1**2(2), 110–112. (See p. 118)

Üstün, B., Melssen, W., and Buydens, L. (2006). Facilitating the application of Support Vector Regression by using a universal Pearson VII function based kernel. *Chemometrics and Intelligent Laboratory Systems*, **8**1(1), 29–40. (See p. 15)

Van Hulse, J., Khoshgoftaar, T. M., and Napolitano, A. (2007). Experimental Perspectives on Learning from Imbalanced Data. In *Proceedings of the 24th International Conference on Machine Learning*, pages 935–942, New York, NY, USA. (See p. 14)

Van Landeghem, S., Saeys, Y., De Baets, B., and Van de Peer, Y. (2008). Extracting protein-protein interactions from text using rich feature vectors and feature selection. In *Proceedings of Third International Symposium on Semantic Mining in Biomedicine*, pages 77–84. (See pp. 30, 31, 33, and 177)

Van Landeghem, S., Ginter, F., Van de Peer, Y., and Salakoski, T. (2011). EVEX: A PubMed-Scale Resource for Homology-Based Generalization of Text Mining Predictions. In *Proceedings of BioNLP 2011 Workshop*, pages 28–37, Portland, OR, USA. (See p. 78)

Van Rijsbergen, C. J. (1979). *Information Retrieval*. Butterworth, London, 2nd edition. (See p. 19)

Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory*. Springer-Verlag New York, Inc. (See pp. 13 and 16)

Veropoulos, K., Campbell, C., and Cristianini, N. (1999). Controlling the Sensitivity of Support Vector Machines. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 55–60. (See p. 15)

Vishwanathan, S. V. N. and Smola, A. J. (2002). Fast Kernels for String and Tree Matching. In *Proceedings of Advances in Neural Information Processing Systems*, pages 569–576. (See pp. 16, 18, and 49)

Wang, B., Spencer, B., Ling, C. X., and Zhang, H. (2008). Semi-supervised Self-training for Sentence Subjectivity Classification. In *Proceedings of the 21st Conference of the Canadian Society for Computational Studies of Intelligence*, pages 344–355. (See p. 68)

Wang, Y. (2010). *Developing Robust Protein Name Recognizers Based on a Comparative Analysis of Protein Annotations in Different Corpora*. Ph.D. thesis, University of Tokyo, Japan. (See pp. 22 and 67)

Weiss, G. M. and Provost, F. (2001). The Effect of Class Distribution on Classifier Learning: An Empirical Study. Technical Report ML-TR-44, Department of Computer Science, Rutgers University. (See p. 14)

Wiegers, T. C., Davis, A. P., and Mattingly, C. J. (2014). Web services-based text-mining demonstrates broad impacts for interoperability and process simplification. *Database (Oxford)*, **2**014. (See p. 23)

Wolpert, D. H. (1992). Stacked Generalization. *Neural Networks*, **5**, 241–259. (See p. 51)

Wright, A., Chen, E. S., and Maloney, F. L. (2010). An automated technique for identifying associations between medications, laboratory results and problems. *Journal of Biomedical Informatics*, **4**3(6), 891–901. (See p. 23)

Xu, R. and Wang, Q. (2013). Large-scale extraction of accurate drug-disease treatment pairs from biomedical literature for drug repurposing. *BMC Bioinformatics*, **1**4, 181. (See p. 24)

Yakushiji, A., Miyao, Y., Tateisi, Y., and Tsujii, J. (2005). Biomedical Information Extraction with Predicate-Argument Structure Patterns. In *Proceedings of the first International Symposium on Semantic Mining in Biomedicine*, Hinxton, UK. (See pp. 30 and 31)

Yakushiji, A., Miyao, Y., Ohta, T., Tateisi, Y., and Tsujii, J. (2006). Automatic Construction of Predicate-argument Structure Patterns for Biomedical Information Extraction. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 284–292, Sydney, Australia. (See p. 30)

Yang, Z., Tang, N., Zhang, X., Lin, H., Li, Y., and Yang, Z. (2011). Multiple kernel learning in protein-protein interaction extraction from biomedical literature. *Artificial Intelligence in Medicine*, **5**1(3), 163–173. (See pp. 31 and 36)

Yao, L., Riedel, S., and McCallum, A. (2010). Collective Cross-Document Relation Extraction Without Labelled Data. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1013–1023, Cambridge, MA, USA. (See pp. 111 and 112)

Yepes, A. J. and Verspoor, K. (2014). Mutation extraction tools can be combined for robust recognition of genetic variants in the literature. *F1000 Research*, **3**, 18. (See p. 118)

Yip, Y. L., Lachenal, N., Pillet, V., and Veuthey, A.-L. (2007). Retrieving mutation-specific information for human proteins in uniprot/swiss-prot knowledgebase. *Journal of Bioinformatics and Computational Biology*, **5**(6), 1215–1231. (See p. 3)

Zeeberg, B. R., Riss, J., Kane, D. W., Bussey, K. J., Uchio, E., Linehan, W. M., Barrett, J. C., and Weinstein, J. N. (2004). Mistaken identifiers: gene name errors can be introduced inadvertently when using Excel in bioinformatics. *BMC Bioinformatics*, **5**, 80. (See p. 22)

Zhang, E. E. and Kay, S. A. (2010). Clocks not winding down: unravelling circadian networks. *Nature Reviews. Molecular Cell Biology*, **1**1(11), 764–776. (See p. 134)

Zhang, H., Huang, M., and Zhu, X. (2011a). Protein-protein interaction extraction from bio-literature with compact features and data sampling strategy. In *Proceedings of the 4th International Conference on Biomedical Engineering and Informatics*, pages 1767–1771. (See p. 30)

Zhang, M., Zhou, G., and Aw, A. (2008). Exploring Syntactic Structured Features over Parse Trees for Relation Extraction Using Kernel Methods. *Information Processing & Management*, **4**4(2), 687–701. (See p. 36)

Zhang, Y., Lin, H., Yang, Z., and Li, Y. (2011b). Neighborhood hash graph kernel for protein-protein interaction extraction. *Journal of Biomedical Informatics*, **4**4(6), 1086–1092. (See pp. 30 and 35)

Zhang, Y., Lin, H., Yang, Z., Wang, J., and Li, Y. (2012). Hash subgraph pairwise kernel for protein-protein interaction extraction. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, **9**(4), 1190–1202. (See p. 31)

Zhu, X. (2008). Semi-Supervised Learning Literature Survey. Technical Report 1530, University of Wisconsin-Madison. (See pp. 32 and 67)

Zwart-van Rijkom, J. E. F., Uijtendaal, E. V., ten Berg, M. J., van Solinge, W. W., and Egberts, A. C. G. (2009). Frequency and nature of drug-drug interactions in a Dutch university hospital. *British Journal of Clinical Pharmacology*, **6**8(2), 187–193. (See p. 45)

Zweigenbaum, P., Demner-Fushman, D., Yu, H., and Cohen, K. B. (2007). Frontiers of biomedical text mining: current progress. *Briefings in Bioinformatics*, **8**(5), 358–375. (See p. 3)

List of Figures

List of Tables

Selbständigkeitserklärung

Hiermit erkläre ich,

- dass ich die vorleigende Arbeit selbstständig und nur mit Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe,
- dass ich keinen Doktorgrad im Fach Informatik besitze,
- und dass mir die Promotionsordnung der Mathematisch-Naturwissenschaftlichen Fakultät der Humboldt Universität zu Berlin vom 17.01.2005, zuletzt geändert am 13.02.2006, veröffentlicht im amtlichen Mitteilungsblatt Nr. 34/2006, bekannt ist

Berlin, den 24. November 2015

Philippe Thomas